

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/DSS/ENS/97J-4

COMPUTER-BASED METHODS FOR CONSTRUCTING TWO-LEVEL
FRACTIONAL FACTORIAL EXPERIMENTAL DESIGNS WITH A
REQUIREMENT SET

DISSERTATION

Steven L. Forsythe, Major, USAF

AFIT/DSS/ENS/97J-4

Approved for public release; distribution unlimited

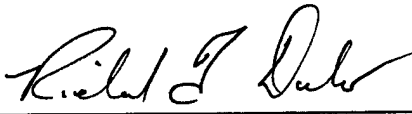
20011115 156


The views expressed in this dissertation are those of the author and do not reflect
the official policy or position of the United States Air Force,
Department of Defense or the U. S. Government.

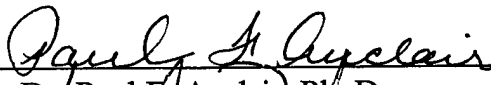
COMPUTER-BASED METHODS FOR CONSTRUCTING TWO-LEVEL
FRACTIONAL FACTORIAL EXPERIMENTAL DESIGNS WITH A
REQUIREMENT SET

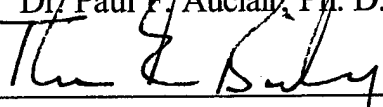
Steven L. Forsythe, B.S., M.S.
Major, USAF

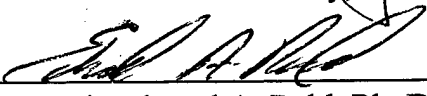
Approved:

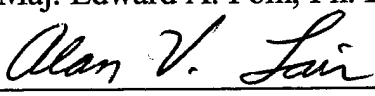
Committee Chairman: 
Dr. Richard F. Deckro, D.B.A.

Committee Member: 
Dr. Kenneth W. Bauer, Ph. D.

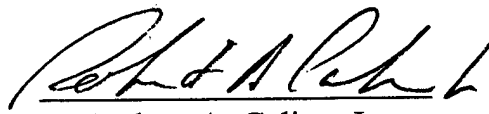
Committee Member: 
Dr. Paul F. Auclair, Ph. D.

Committee Member: 
Dr. Thomas G. Bailey, Ph. D.

Committee Member: 
Maj. Edward A. Pohl, Ph. D.

Dean's Representative: 
Dr. Alan V. Lair, Ph. D.

Accepted:


Dr. Robert A. Calico, Jr.
Dean, School of Engineering

Acknowledgments

It is impossible to adequately thank all those who have contributed to this dissertation. Some of the most significant persons are mentioned below, but I extend a general thank you to all those students, professors, coworkers, family, and friends who encouraged and aided me in this effort.

I thank my advisor, Dr. Richard F. Deckro, and my committee members, Dr. Kenneth W. Bauer, Lt. Col. Thomas G. Bailey, Dr. Paul F. Auclair, and Maj. Edward A. Pohl, for their perseverance, understanding, and assistance.

I thank Lt. Col. David Coulliette and Dr. James S. Shedden for their support. Col. Kurt Cichowski's encouragement was pivotal in the completion of this dissertation. I would like to thank all of my coworkers and leadership at Air Force Studies and Analyses Agency for their encouragement and assistance.

My greatest thanks go to my wife, Sandy. Her love, encouragement, and editing efforts were vitally needed and deeply appreciated.

This dissertation is dedicated to my parents for the great job they've done throughout the years. As a parent, I gain new appreciation for the trials and tribulations the seven of us kids put them through.

Steven L. Forsythe

Table of Contents

Acknowledgments	iv
Table of Contents.....	v
List of Tables	x
List of Figures	xii
Abstract.....	xiii
I. Introduction.....	1-1
II. DOE Background	2-1
2.1 Introduction	2-1
2.2 DOE Development.....	2-2
2.3 Fractional Factorial Designs.....	2-3
2.3.1 The DOE/RS Problem	2-5
2.4 Manual Methods for Solving the DOE/RS Problem.....	2-6
2.4.1 Taguchi Graphs.....	2-7
2.4.2 Other Graphical Methods.....	2-9
2.4.3 Confounding Tables	2-12
2.4.4 Defining Relations.....	2-13
2.5 Complexity of the DOE/RS Problem	2-16
2.6 Alternatives to Optimizing the Fractional Factorial Designs	2-18
2.7 Research Approach	2-19

III. Optimal Methodology.....	3-1
3.1 Introduction.....	3-1
3.2 Mathematical Model of DOE with Requirement Set (DOE/RS).....	3-1
3.2.1 Objective Function.....	3-2
3.2.2 IP Formulation Constraints.....	3-8
3.2.2.1 Assignment Constraints.....	3-8
3.2.2.2 Resolution III Constraints.....	3-8
3.2.2.3 Minimum Aberration Constraint.....	3-9
3.2.2.4 Confounding Constraints.....	3-10
3.2.2.5 Interaction Constraints.....	3-11
3.2.3 Complete IP Formulation.....	3-13
3.2.4 Example Problem.....	3-14
3.2.4.1 Objective Function.....	3-15
3.2.4.2 Assignment Constraints.....	3-15
3.2.4.3 Resolution III Constraints.....	3-16
3.2.4.4 Confounding Constraints.....	3-16
3.2.4.5 Interaction Constraints.....	3-17
3.2.4.6 Minimum Aberration Constraint.....	3-18
3.2.4.7 Integer Constraints.....	3-18
3.2.4.8 Optimal Solution.....	3-18
3.3 The IP Experiment.....	3-19
3.3.1 The Baseline Experimental Design.....	3-21
3.3.2 IP Results.....	3-22

3.3.2.1 Specific Solutions	3-26
3.3.2.1.1 Solution to Problem One	3-26
3.3.2.1.2 Solution to Problem Two	3-28
3.4 Summary	3-28
IV. Heuristic Methodologies	4-1
4.1 Introduction	4-1
4.2 Heuristics	4-1
4.2.1 Evaluation of Heuristics	4-2
4.3 CAVE/ICA VE Heuristics	4-3
4.3.1 CAVE	4-4
4.3.1.1 Solving an Example Problem Using CAVE	4-6
4.3.2 ICA VE	4-8
4.3.2.1 Solving an Example Problem Using ICA VE	4-11
4.3.3 Characteristics of the CAVE and ICA VE Heuristics	4-12
4.3.4 The CAVE/ICA VE Experiment	4-12
4.3.5 The CAVE/ICA VE Results	4-13
4.4 Simulated Annealing	4-15
4.4.1 SA Applications	4-17
4.4.2 Characteristics of SA	4-18
4.4.3 SA Structure	4-19
4.4.3.1 Problem Depiction	4-20
4.4.3.2 Perturbation Mechanism	4-20
4.4.3.3 Cooling Schedule	4-21

4.4.3.3.1 Starting Temperature.....	4-22
4.4.3.3.2 Temperature Length.....	4-23
4.4.3.3.3 Cooling Rate.....	4-24
4.4.3.4 Acceptance Function.....	4-25
4.4.3.5 Stopping Criteria.....	4-26
4.4.3.6 An Example Problem.....	4-28
4.4.4 The SA Parameter Experiment	4-29
4.4.5 The Parameter Experimental Result.....	4-31
4.4.6 The DESA Evaluation Experimental Result.....	4-34
4.5 Summary	4-39
V. Conclusions.....	5-1
5.1 Summary.....	5-1
5.2 Comparative Results	5-2
5.2.1 Performance of the Solution Methodologies.....	5-2
5.2.2 Computational Effort of the Solution Methodologies.....	5-2
5.2.3 Efficiencies Achieved by Solving DOE/RS Problems.....	5-4
5.3 Recommendations for Future Research.....	5-5
5.4 Conclusion	5-6
VI. Bibliography.....	Bib-1
VII. Appendices	Apx-1
A. Experimental Results from DESA Experiments	Apx-1
B. Experimental Results from ICAVE Heuristic Experiments	Apx-9
C. Bisgaard and Fuller Example Problem.....	Apx-11

D. Codes Used to Write the 0/1 IP Problem File	Apx-33
E. Code Used to Implement the ICAVE Heuristic.....	Apx-43
F. Codes Used to Implement the SA Heuristic	Apx-52
G. Cost and Value Functions	Apx-63
H. DOE Nomenclature.....	Apx-65
Vita	Vita-1

List of Tables

Table 2-1 A Confounding Table	2-12
Table 2-2 A 2^3 Full Factorial Design.....	2-13
Table 2-3 Maximum Resolution, Minimum Aberration Designs	2-15
Table 3-1 DOE/RS Problem Terms.....	3-3
Table 3-2 2^{2-0} Experimental Design.....	3-15
Table 3-3 Experimental Factors.....	3-20
Table 3-4 Dependent Variables.....	3-21
Table 3-5 The Baseline Experimental Design.....	3-22
Table 3-6 The Results of the IP Methodology.....	3-26
Table 3-7 IP Solution to Problem One (16/12).....	3-27
Table 4-1 The Overall CAVE Performance.....	4-13
Table 4-2 The Overall ICAVE Performance.....	4-14
Table 4-3 CAVE vs. ICAVE Performance.....	4-15
Table 4-4 Overall SA Design	4-31
Table 4-5 The Overall DESA Performance.....	4-36
Table 4-6 The Median DESA Performance	4-36
Table 5-1 Comparison of DOE/RS Design to R_v Design.....	5-4
Table A-1 Results of DESA Experiments.....	Apx-2
Table B-1 ICAVE Results for Runs with Zero Deviations.....	Apx-9
Table B-2 ICAVE Results for Runs with Positive Deviations	Apx-10
Table H-1 An 8-Trial Full Factorial Design.....	Apx-65

Table H-2 A 16-Trial Experimental Design.....	Apx-66
Table H-3 A 32-Trial Experimental Design.....	Apx-67
Table H-4 A 64-Trial Experimental Design.....	Apx-68

List of Figures

Figure 2-1 A Small DOE/RS Requirement Set.....	2-7
Figure 2-2 Two Taguchi Graphs for an 8-Run Experiment.....	2-8
Figure 2-3 RS Graph and Matching Design.....	2-10
Figure 3-1 Number of Variables vs. Experiment.....	3-23
Figure 3-2 Number of Constraints vs. Number of Experimental Design Points	3-24
Figure 4-1 Pseudocode for CAVE Heuristic	4-4
Figure 4-2 Pseudocode for ICAGE Heuristic.....	4-9
Figure 4-3 Functional Evaluations Required by ICAGE in Solving DOE/RS Problem 32/25.....	4-11
Figure 4-4 Pseudocode for a Simulated Annealing Algorithm.....	4-19
Figure 4-5 The Effect of Temperature on SA Behavior.....	4-26
Figure 4-6 The Effect of SA Parameters on Solution Quality.....	4-32
Figure 4-7 The Effect of SA Parameters on Computational Effort	4-34
Figure 4-8 Histogram of DESA Effectiveness.....	4-37
Figure 4-9 Performance vs. Evaluations	4-38
Figure 5-1 Mean Computational Effort vs. Percentage of Relative Reduction For the Four Methodologies Under Study.....	5-3

ABSTRACT

This dissertation developed four methodologies for computer-aided experimental design of two-level fractional factorial designs with requirement sets (DOE/RS). The requirement sets identify all the experimental factors and the appropriate interaction terms to be evaluated in the experiment.

Taguchi graphs and similar manual methods provide techniques for solving the DOE/RS problem. Unfortunately, these methods are limited because they become difficult to use as the number of factors or interaction terms exceeds ten. This research showed that the DOE/RS problem belongs to a class of difficult-to-solve problems known as NP-Complete. It is the combinatorial nature of NP-Complete problems that causes them to become computationally challenging as the size of the problem increases. Heuristics are often used to find good solutions to large NP-Complete problems.

The methodologies developed in this dissertation have been used to solve DOE/RS problems involving up to 16 factors and 47 interaction terms, thus significantly increasing the upper limit of tractable DOE/RS problems over current methods. A 0-1 integer programming formulation and three heuristic methods were developed and evaluated in their ability to solve the DOE/RS problem.

The 0-1 IP formulation was developed as an algorithmic approach. The IP formulation provides a standard optimization methodology that guarantees a global optimum solution. Due to the NP-Complete nature of the DOE/RS problem, however,

the IP formulation became computationally intractable as the size of the problem increased.

Two new heuristics, Column Assignment Via Examination (CAVE) and Iterative Column Assignment Via Examination (ICAVE), were specifically designed to take advantage of the special structure of the fractional factorial design. The CAVE heuristic quickly generates feasible designs that can serve as starting points for other solution techniques. The ICAVE heuristic performance averaged a 99% solution quality.

A simulated annealing approach was used as the optimization strategy in the Design of Experiments via Simulated Annealing (DESA) heuristic. DESA performed consistently over a range of SA parameters. Recommended parameter settings were identified. DESA provided a standard heuristic approach as a baseline for comparison against the other solution methodologies applied to the DOE/RS problem.

ICAVE provided higher quality solutions at the expense of greater computational effort when compared to the CAVE and DESA heuristics. For smaller DOE/RS problems, the IP formulation generates an optimal solution. An experimenter should choose the solution methodology that best matches the needs of his or her experiment, expertise, and available resources.

I. Introduction

Experimental design is a key factor in today's ever more complex and competitive world. Whether one is discussing critical scientific experimentation, basic research, developmental engineering or commercial product design and production, effective experimental design is a bulwark of efficient, competitive practices (Hockman and Jenkins, 1994, p. 50). Schmidt and Launsby (1991, preface) highlighted the importance of experimental design with a short discussion on the growing necessity of developing efficient experimental designs for research and manufacturing:

The time is quickly approaching when you will not be considered a competent engineer or technical manager without a working knowledge of Experimental Designs. The demands of increased efficiency of process, lower product cost, and shortened development cycles will dictate that we use simple, but powerful tools to get the most out of our experiments. No longer do we have the luxury of running one-factor-at-a-time experiments or experiments with excessive sample sizes. In competitive environments, only those groups that **apply** experimental design approaches efficiently and effectively will survive. (Schmidt and Launsby 1991, preface)

The effective use of experimental resources is profoundly affected by the design choice (Shannon 1975, p. 144). Without an efficient and effective design, answering the experimental question may be possible only through prohibitively excessive expenditures of time and resources. In our globally competitive world of ever decreasing time to market and contracting budgets, time and resources are often in short supply.

A full factorial design is a simple, although potentially inefficient, experimental design. A full factorial design requires one experiment for every combination of settings or levels among

the experimental variables. The size of such experiments grows prohibitively as the number of experimental variables increases. When the size of the experiment becomes too large, an alternative is to construct a fractional factorial design (FFD), utilizing a critical subset of experiments.

Using a FFD to reduce the size of the experiment results in increased experimental efficiency, provided the experimental design chosen still yields the information needed by the experimenter. Several methodologies have been developed to aid the experimenter in selecting fractional factorial designs (Franklin 1985, Box and Draper 1987, Taguchi and Koushi 1987, Lee 1991, Wu and Chen 1992). While these techniques are effective for small designs, selecting the “best” FFD for large experiments is difficult in all but the simplest cases with the currently available techniques. The “best” design, defined precisely in Chapter Three, is essentially the most cost and time effective design that provides the necessary information about the experimental variables of interest. As Mandu and Kuei (1992, p. 97) point out, “When dealing with a large number of controllable factors, the construction of fractional factorial designs becomes extremely difficult.” The central challenge in developing a design is to strike a critical balance between obtaining the necessary information and the size (and therefore, cost in dollars, time, or effort) of the experiment.

This dissertation develops a computer-aided design methodology that will assist the experimenter in seeking an effective experimental design. In addition, the research provides a method for evaluating the relative importance of various possible tradeoffs among competing designs when the specified ideal criteria cannot be satisfied. These optimization procedures extend design capabilities that are currently applicable only to small experiments.

A zero-one integer programming (0-1 IP) approach, as well as three heuristics for executing a computerized search for an optimum design, are developed in this research. These automated procedures save the experimenter's valuable design phase time by searching for efficient designs in a timely manner and reducing the manual mechanics of constructing experimental designs. An experimenter may use the computerized design system to judiciously create a portfolio of candidate designs, possibly with different design criteria, and then select from this portfolio the design that best fulfills the requirements of the experiment. The approach develops a base design that the experimenter may then evaluate and adopt or adjust to incorporate his or her subjective, non-quantifiable, knowledge, experience, and judgment.

There are three principal objectives of this research. The first objective is to develop a set of effective methods for solving Design of Experiments with a Requirement Set (DOE/RS) problems. The DOE/RS problem is to select the best fractional factorial design when the experimenter has defined a set of terms to be estimated. The DOE/RS problem is fully defined and discussed in Chapter Two. The second objective of this research is to demonstrate that the proposed solution methods effectively solve DOE/RS problems for experiments that are currently too large for available methods. The third objective is to evaluate the effectiveness of the solution techniques.

Chapter Two introduces Design of Experiments (DOE) and provides a review of the current state-of-the-art in developing fractional factorial designs. Chapter Two also provides the Design of Experiments with Requirement Set problem formulation.

Chapter Three builds on the body of knowledge discussed in Chapter Two and develops a 0-1 integer programming methodology to solve the DOE/RS problem. This chapter

also presents an experiment to evaluate and characterize the performance of the IP to solve the DOE/RS problem.

Chapter Four introduces three heuristic methods as alternative solution methodologies to the 0-1 IP. Two new heuristics, Column Assignment Via Examination (CAVE) and Iterative Column Assignment Via Examination (ICAVE), were specifically designed to take advantage of the special structure of the fractional factorial design. The CAVE heuristic quickly generates feasible designs that can serve as starting points for other solution techniques. A simulated annealing approach was used as a third heuristic optimization strategy in the Design of Experiments via Simulated Annealing (DESA) heuristic. DESA provided a general-purpose heuristic benchmark to compare with the performance of the CAVE and ICAVE heuristics. In each case, an experiment was conducted to characterize the performance of each heuristic. The results of these experiments are provided and discussed in this chapter. The complete experimental data is provided in the appendices.

Finally, Chapter Five compares the various DOE/RS methodologies, summarizes the contributions made by this research, and suggests fruitful paths to follow as extensions to it.

II. DOE Background

This chapter provides background on Design of Experiments (DOE): specifically, information about fractional factorial designs, the DOE with a Requirement Set (DOE/RS) problem, and a review of current methodologies for solving the DOE/RS problem. Readers who desire additional background on DOE are directed to the following references: Box, Hunter, and Hunter (1978), Taguchi and Wu (1980), Hicks (1982), Box and Draper (1987), Taguchi and Konishi (1987), Schmitt and Launsby (1991), Montgomery and Myers (1995), Montgomery (1999), and Mee and Peralta (2000).

2.1 Introduction

Designed experiments seek to provide the information necessary to prove or disprove a hypothesis. Clearly, a well-structured experiment utilizes the fewest resources necessary to test the proposed hypothesis to the level of significance required. Once an experiment is completed and the data analyzed, new insights and information may lead to the formation of another hypothesis and further experimentation. Box and Draper (1987) refer to such a process as sequential experimentation.

The combination of main effects and key interactions of interest compose the *requirement set* (RS) of the experiment (Greenfield 1976, p. 64). The RS identifies the size and nature of the experiment required to meet the experimenter's objectives. Ideally, an experiment allows the unambiguous estimation of each term in the RS while minimizing the experimental trials, since unnecessary trials represent wasted resources.

2.2 DOE Development

Two-level experimental designs are often used to collect data on the linear effects of experimental variables (Nachtsheim 1987, p. 147). A *full factorial design* is an experiment that uses every combination of specified levels of experimental variables, allowing the estimation of each variable and all possible interaction terms. The experimental levels for each variable can be either the actual experimental values or coded values such as “+1” or “-1” values. The “+1” value represents the “high” setting of an experimental factor and the “-1” represents the “low” setting. Using coded values enables the design to be balanced and orthogonal. If the design region is sufficiently small, a polynomial function with linear terms and second-order interaction terms will often suffice to model the process under study (Box and Draper 1987, p. 3).

Unfortunately, full factorial designs very rapidly become prohibitively large. If K variables are to be studied, a two-level full factorial design requires 2^K experiments. If the experiments to be conducted are relatively inexpensive in time and cost when compared to the value of the information to be gained, a full factorial design may be desirable. Budget or time limitations may restrict the experimenter's options. In addition, often only the first and some second-order terms are found to be significant (Box and Draper 1987, p. 21). When a full factorial design is deemed too costly or lengthy for its potential benefits, a fraction of the original experiment may present a suitable alternative. The reduced experimental design is referred to as a *fractional factorial design* (FFD). If analysis of the experimental data from the initial fractional factorial design experiment indicates that additional data is required or desired, subsequent experiments can be added.

2.3 Fractional Factorial Designs

Fractional factorial designs are denoted as: N^{K-P} . N in this context is the number of levels; typically two- or three-level designs are used. K indicates the number of experimental variables. The numerical evaluation of N^{K-P} indicates the number of experimental trials. For example, a 2^{6-1} experiment is a two-level, 32-run experiment. Experiments where P equals one have a special name: they are called one-half fraction designs since they utilize one-half of the trials in a full factorial design with the same number of experimental variables.

Fractional factorial designs support the following types of experimental investigations:

(1) Research where a large number of main effects can be “screened” for potential significance, thus identifying the most significant effects for further study.

(2) Research where a set of interactions can reasonably be assumed to be negligible. Effects of third-order and higher are often assumed to be negligible.

(3) Iterative experimentation where groups of trials are to be performed based on results of prior trials (Fries and Hunter 1980, p. 601).

In a fractional factorial design, two terms are *aliased* or confounded if effects due to each of the two terms are impossible to distinguish from each other. The aliasing or confounding structure of a FFD is determined by its defining relation, which is simply the set of columns in the design matrix consisting of all positive 1 or negative 1 values for two-level designs. The *resolution* of a design equals the length of the shortest word (column name) in its defining relation. The resolution defines the degree of aliasing among design factors. For example, a Resolution III design has at least one first-order term aliased with one second-order term. In a Resolution V design, second-order terms are not confounded with terms which are less than

third-order, and first-order terms are not confounded with terms which are less than fourth-order.

Box and Draper (1987, p. 154) state that the higher order terms are typically assumed to be insignificant, since the response of most physical systems can be modeled as a low-order polynomial of the input variables for a small experimental design region. A model which contains only first and second-order terms can always be estimated using a Resolution V design. These second-order terms represent the product of two experimental variables and are referred to as two-factor interactions or *2fi*'s.

Resolution does not, however, account for the "amount" of aliasing in a design; it is only a measure of the degree of aliasing. For example, a Resolution IV (R_{IV}) design with only one four-letter word in its defining relation would have fewer *2fi*'s confounded with other *2fi*'s, relative to a similar R_{IV} design with four four-letter words in the defining relation.

Fries and Hunter (1980, p. 601) extended the concept of resolution to include minimum aberration. The *aberration* of a design is the number of minimum length words in the defining relation. Fries and Hunter state that for a given number of trials, all fractional factorial designs of maximum resolution are not equal. When comparing two fractional factorial designs with equal resolution, Fries and Hunter assert that the design with the lower aberration is a superior design. A design with the smallest possible aberration is termed a *minimum aberration design*.

For example, one possible 2^{9-3} design assigns the last three factors to the following columns: $G=ABC$, $H=ADE$, and $J=BEF$, and is a R_{IV} design. This design has the defining relations $I = ABCG = ADEH = BEFJ = BCDEGH = ACEFGJ = ABDFHJ = CDFGHJ$. This design has three four-letter words in the defining relation and has an aberration of three.

Another possible $2^{9-3} R_{IV}$ design can be defined by the following factor assignments: $G=ABCD$, $H=ACEF$, and $J=CDEF$. This design has the defining relations $I = ABCDG = ACEFH = CDEFJ = BDEFGH = ABEFGJ = ADHJ = BCGHJ$ which has only one four-letter word in the defining relation. The second design has an aberration of one. As a result, the number of $2fi$'s confounded with other $2fi$'s is much smaller in the second design even though both are R_{IV} designs. The second design is superior to the first design in the number of $2fi$'s which can be clearly estimated (assuming higher order terms are negligible). The second design is, in fact, a minimum aberration design since no design can have an aberration smaller than 1 (i.e., there must always be at least one smallest term in the defining relations of a fractional factorial design).

Maximizing the resolution and minimizing the aberration of a design both assume all n -factor interactions are equally important. Fries and Hunter implicitly assumed there is no prior knowledge gained from past experimentation or expert knowledge that might suggest a grouping of two-factor interactions based on their potential significance.

2.3.1 The DOE/RS Problem. Selecting a fractional factorial design when a requirement set is specified is defined in this research as the Design of Experiments with a Requirement Set (DOE/RS) problem. The problem assumes that the experimenter desires to select the smallest experiment that will provide clear estimates of each term in the requirement set (RS). Unnecessarily large experiments waste resources, while experiments which are too small, or have confounding among the terms in the RS, fail to provide unambiguous information. The goal in solving the DOE/RS problem is to strike the correct balance between the size (and therefore,

cost in dollars, time, or effort) of the experiment and the value of the information gained from the experiment.

2.4 Manual Methods for Solving the DOE/RS Problem

DOE/RS problems appear in experimental designs associated with diverse situations such as manufacturing, engineering, and simulation (Taguchi and Konishi, 1987). A variety of manual methods can solve small DOE/RS problems. Current manual methods are reviewed in this section. The following DOE/RS problem is used to illustrate the manual methods.

Consider an eight-run experiment utilizing a 2^{4-1} fractional factorial design. The columns of the design matrix are: the mean, A, B, AB, C, AC, BC, and ABC. For this example, the requirement set consists of {mean, a, b, c, d, ab, ac, and ad}.

Assigning the variable d to the ABC column in the design matrix ($d \rightarrow ABC$), forces $ad \rightarrow BC$. If the remaining terms are assigned $a \rightarrow A$, $b \rightarrow B$, and $c \rightarrow C$, then $ab \rightarrow AB$, and $ac \rightarrow AC$. Larger problems are significantly more difficult to solve using manual methods. Consider a DOE/RS problem that is a 2^{11-5} two-level fractional factorial design with 44 two-factor interaction terms in the requirement set. The requirement set is {a, b, c, d, e, f, g, h, i, j, k, and the 2fi's ab, ac, ad, ae, af, ag, ah, ai, aj, ak, bc, bd, be, bg, bh, cd, ce, cf, cg, ch, ci, cj, ck, df, dh, di, dj, dk, ef, eh, ei, ej, ek, fg, fh, fj, fk, gh, gi, gj, gk, hi, hj, hk}. Since each first-order term can be assigned to any column, there are $11!$ (39,916,800) combinations that match the 11 experimental variables to the 11 factors in the Resolution III experimental design. If a new possible design could be examined each second, it would require more than 11,000 hours to examine all possible designs.

Several manual methods exist to aid the experimenter in choosing an experimental design. The most well known and widely used of the manual approaches is a methodology developed by Taguchi and Konishi (1987), which is presented in the next subsection.

2.4.1 Taguchi Graphs. Taguchi's linear graph method is visually appealing and straightforward to apply to small DOE/RS problems. The method utilizes "point-line" graphs, where the vertices represent main effects and their connecting line segments signify two-factor interactions (2fi's) (Taguchi and Konishi 1987, Wu and Chen 1992). Selecting a design begins with creating such a graph for the requirement set. Figure 2-1 shows the RS depicted in graphical form. This requirement set graph is then compared to Taguchi's catalog of graphs for a particular run size.

Figure 2-2 shows the two possible Taguchi graphs for a two-level design of eight trials. Compare the RS in Figure 2-1 with the two designs in Figure 2-2. One can easily see that the first design matches the RS presented here: {a, b, c, d, ab, ac, ad} and is the appropriate choice. The variable a is assigned to the node (the "1", representing the first column in the design matrix) with three lines or arcs connecting it to the other three nodes. The other three nodes are assigned to the variables b, c, and d respectively. The arcs indicate that the ab, ac,

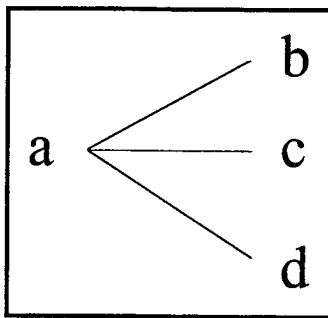


Figure 2-1. A Small DOE/RS Requirement Set

and ad terms can be estimated. When the requirement set graph matches one of the listed graphs, factor names are assigned to corresponding columns in the design matrix (Taguchi and Konishi, 1987). The numbers identify columns in the design matrix, A=1,

$B=2, AB=3, C=4, AC=5, BC=6, ABC=7, D=8, \dots$). The resulting design permits clear estimation of all members of the requirement set; $a \rightarrow 1$ (i.e. A), $b \rightarrow 2$ (i.e. B), $c \rightarrow 4$ (i.e. C), $d \rightarrow 7$ (i.e. ABC).

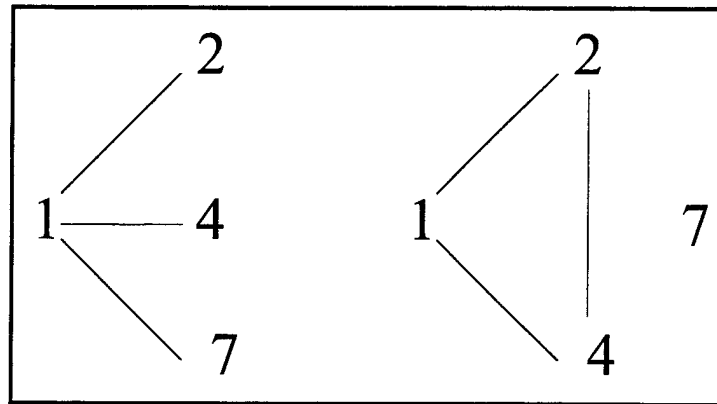


Figure 2-2. Two Taguchi Graphs for an 8-Run Experiment

Unfortunately, there are a number of limitations to Taguchi's method. Schmidt and Launsby (1991, pp. 3-33) state that Taguchi's designs are not new, but are simply forms of Plackett-Burman, Fractional Factorial, or Latin Square designs. Wu and Chen (1992) showed that (1) the designs for all but two of Taguchi's graphs are for Resolution III; (2) better graphs in terms of minimum aberration are possible; and (3) Taguchi's collection of graphs is incomplete. For example, only six graphs are available for developing 16-run designs, although more than 800 such minimum aberration designs are possible. For other two-level fractional factorial designs (2^{6-2} , 2^{9-5} , and 2^{12-7}), no graphs are provided (Taguchi and Konishi 1987, p. 162 - 165). With such limited coverage, Taguchi's method could ignore several potentially superior designs.

Section 2.5 presents discussion and a proof showing that the problem of identifying a subgraph of the Taguchi graph, that is isomorphic to the RS, is an NP-Complete problem. One graph is isomorphic to a second graph if relabeling the nodes of the first graph can produce a graph that is identical to the second graph. Since the DOE/RS problem can be considered as solving a set of Taguchi graph problems, the DOE/RS is NP-Complete. Two well-known examples of NP-Complete problems are the Traveling Salesperson Problem (TSP) and the Knapsack Problem. One critical aspect of NP-Complete models is that a large NP-Complete problem may be computationally challenging for an algorithm. Therefore, manual methods, as well as computer-based algorithms, become difficult to solve if the NP-Complete problem is large.

2.4.2 Other Graphical Methods. Given the limitations of Taguchi's linear graphs, several researchers have presented improvements to Taguchi's approach to solving the DOE/RS problem. Wu and Chen (1992) presents a modification of Taguchi's method that overcomes a number of the initial limitations of Taguchi's graphs. Starting with a requirement set graph and a minimum aberration design, Wu's method attempts to find a subgraph of the design graph that is isomorphic to the requirement set graph. If no match is found, the procedure is repeated with the next-best design, where best is given by the minimum aberration criteria.

Figure 2-3 presents two linear graphs to illustrate Wu's method for a small problem. The first graph represents the RS, while the second could have resulted from a search of possible fractional factorial designs to identify one that satisfies the experimental design problem.

The information in Figure 2-3 can be used to create a satisfactory experimental design. A node in the top graph represents each experimental variable. These variables are assigned to

the columns in a design matrix that are indicated by the numbers of the corresponding nodes in the bottom graph.

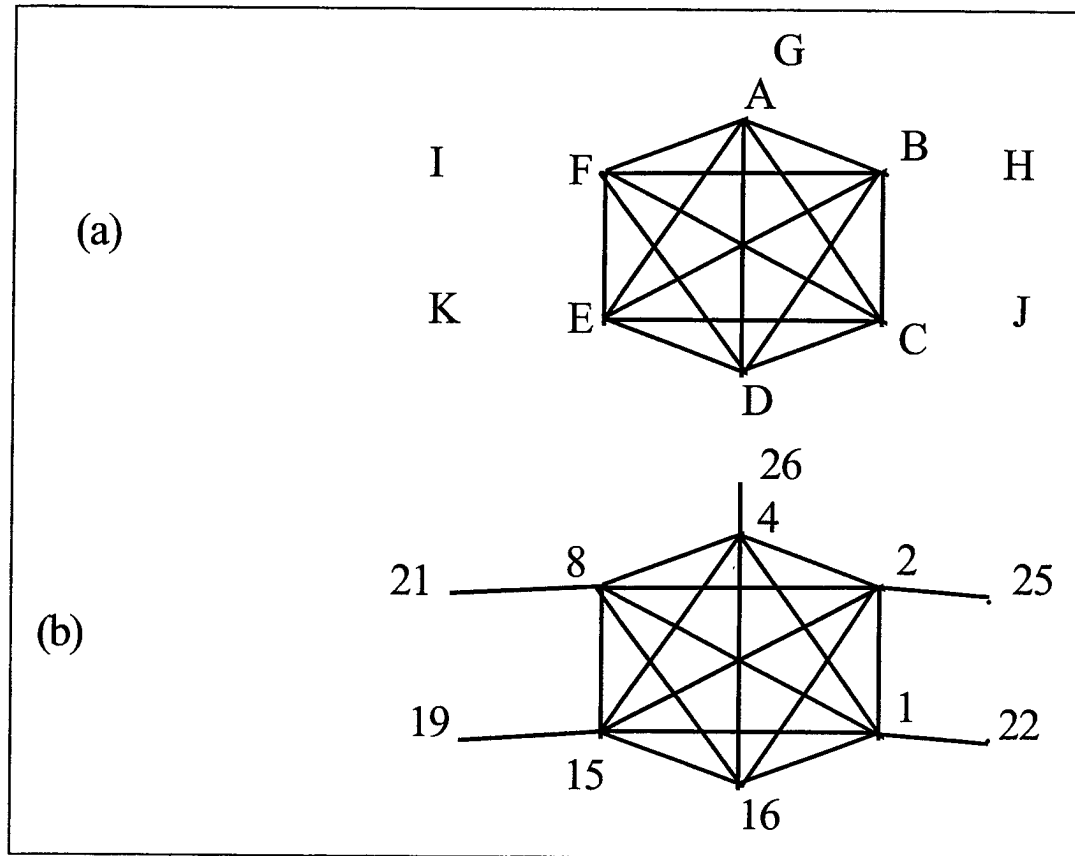


Figure 2-3. RS Graph and Matching Design: (a) Represents the RS; Graph (b) is Found to Match the Requirement Set (Wu and Chen 1992, p. 164)

Since this is a very small DOE/RS problem, the RS had a structure that allowed for relatively easy comparisons between RS and potential design graphs. Many problems may not be so straightforward; as problems become large, the graphs become exceedingly complex and unmanageable. The problem given in Figure 2-3 depicts a RS for an experimental design that contains 11 nodes and 20 arcs. Wu's method identifies all the nonisomorphic (unique) graphs

that result from relabeling factor assignments. In applying this procedure to a 2^{11-5} design, Wu identified the total number of possible designs as 5,242,880. (Wu and Chen 1992, p. 165)

For many practical, operational problems, such a vast number of graphs is too large for enumeration or storage. A more efficient approach is therefore needed for the practical implementation of large, complex experimental design problems. In addition, the linear graph approach and its derivatives are limited to binary prior information (weightings) regarding the 2^k 's. With Wu's method, an interaction is either important or it is not. In practice, there may be some prior knowledge regarding the importance of interaction terms. For instance, sets of interactions might be: 1) those known to be significant, 2) those known to be insignificant, and 3) those whose importance is unknown to the investigator.

The difficulties with Wu's method are two-fold. First, a graph for the size of the experiment to be conducted may not be available. While there may be hundreds or thousands of possible graphs that could be used for a given experiment, the experimenter must choose only one. Wu and others have offered no systematic method for choosing an appropriate graph. Without such a systematic solution method Wu's approach becomes intractable as the number of possible graphs becomes large.

Even when an appropriate graph is chosen, the experimenter must still find a subgraph of the linear graph that satisfies the experimenter's RS. Figure 2-3 presents a requirement set in graphical form as well as a satisfactory experimental design, also presented in graphical form. The symmetrical form of the requirement set in this example would aid in manually evaluating a design graph. Requirement sets need not have any special symmetry or structure and may therefore be much more difficult to analyze visually.

2.4.3 Confounding Tables. Lee (1991) presents another method of manually solving the DOE/RS problem. Lee created *confounding tables* that represent all the two-factor interactions among all the columns of a full factorial design matrix.

Table 2-1 represents a 2^3 design. Each column is identified by the italicized numeral in the first row. The subsequent numbers listed below in each column represent the two-factor interactions that are confounded with whatever variable is assigned to that column. If an experimental variable were assigned to each column of the design matrix, v1 to column 1, v2 to column 2, and so on, then each column contains the six two-factor interactions confounded with the variable identified in the first row. In the case of v3, v1v2, v2v1, v4v7, v5v6, v6v5, and v7v4 would all be indistinguishable from v3. Note that the order of the two-factor interaction terms is irrelevant; v1v2 is equivalent to v2v1. Column one is therefore equivalent to the interaction of the terms assigned to columns 2 and 3, 4 and 5, and 6 and 7.

TABLE 2-1: A CONFOUNDING TABLE

<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>
(A)	(B)	(AB)	(C)	(AC)	(BC)	(ABC)
2x3	1x3	1x2	1x5	1x4	1x7	1x6
3x2	3x1	2x1	2x6	2x7	2x4	2x5
4x5	4x6	4x7	3x7	3x6	3x5	3x4
5x4	5x7	5x6	5x1	4x1	4x2	4x3
6x7	6x4	6x5	6x2	6x3	5x3	5x2
7x6	7x5	7x4	7x3	7x2	7x1	6x1

This method can be useful, but Lee himself admits that as the size of the problem exceeds ten variables or so, his method becomes difficult to use:

The only impractical designs in our collection of graphs are those of the 64-runs from 10 factors to 21 factors with considerably large amount of interactions. By comparing the other graphical approaches, it becomes

clear that it is difficult to make a practical approach for the above cases.
(Lee 1991, p. 79)

Lee's method therefore has three limitations: an appropriate confounding table must be available, the problem should be smaller than 64 trials, and a significant expenditure of the experimenter's time is required if the RS has a large number of factors in interaction terms.

2.4.4 Defining Relations. Graphs and interaction charts are not the only manual methods that could be used to select a design. A fractional factorial design can be constructed by specifying the columns to which the experimental factors are assigned. In a 2^{K-P} design, the first K-P terms are assigned to the first-order columns (e.g. A, B, C and so forth). Table 2-2 shows a 2^3 full factorial design that is the basis of an eight experiment FFD. The *defining relation* is the set of assignments of the other factors to interaction columns.

TABLE 2-2: A 2^3 FULL FACTORIAL DESIGN

A	B	AB	C	AC	BC	ABC	I
+1	+1	+1	+1	+1	+1	+1	+1
-1	+1	-1	+1	-1	+1	-1	+1
+1	-1	-1	+1	+1	-1	-1	+1
-1	-1	+1	+1	-1	-1	+1	+1
+1	+1	+1	-1	-1	-1	-1	+1
-1	+1	-1	-1	+1	-1	+1	+1
+1	-1	-1	-1	-1	+1	+1	+1
-1	-1	+1	-1	+1	+1	-1	+1

For example, in a 2^{4-1} , a one-half fraction design, the d factor could be assigned to the ABC interaction column (e.g. $D = ABC$). The a, b, and c factors are assigned to their respective A, B, and C columns. In this example the defining relation is $D = ABC$.

$D = ABC$ implies that **I** (an Identity column of all ones which is associated with estimating the mean response in DOE) equals ABCD.

The variables are coded +1, and -1. The coded variables are related to the uncoded variables by the following formula:

$$CodedVariable = \frac{Variable - (Min + Max)/2}{(Max - Min)/2} \quad (2-1)$$

Min and *Max* are the minimum and maximum values for the variable. *Variable* is the particular variable value to be coded. The *CodedVariable* is the value appearing in the design matrix (Montgomery 1999, p. 296).

Because a column in a two-level design is composed of only 1's and -1's, any such column multiplied (element by element) by itself is equal to the identity column. If the experimental column associated with AC is $(1,1,-1,-1)^T$ and BC is associated with the same column, $(1,1,-1,-1)^T$, then, the AB interaction term is associated with the column $([1*1], [1*1], [-1*-1], [-1*-1])^T$, which is $(1,1,1,1)^T$. Therefore, the AB estimate is confounded with the estimate of the mean for this experimental design since the mean is always associated with a column of 1's in the experimental design.

Larger designs have more complex defining relations. Suppose the example is expanded by adding an additional term $E = AB$. Therefore, $EE = ABE$. This implies that $\underline{I} = ABE$ and we have already seen that $\underline{I} = ABCD$. Multiplying $ABE*ABCD$ yields another term, CDE, equivalent to \underline{I} . Therefore $\underline{I} = ABE = CDE = ABCD$. To once again identify which terms are confounded with AD, multiply AD and each of the three terms that are equal to the identity column; $AD*ABE = BDE$, $AD*CDE = ACE$, $AD*ABCD = BC$. Therefore, AD is confounded with BC, BDE, and ACE. In order to estimate both the AD term and the BC term in an unambiguous fashion, a different design would need to be selected.

Such man-in-the-loop iterative methods are acceptable for small designs with relatively small requirement sets. The defining relation in a large, highly fractionated design may be complex and difficult to use with this method (Box and Draper, ch. 5).

TABLE 2-3: MAXIMUM RESOLUTION, MINIMUM ABERRATION DESIGNS

	Factors :	Factors:	Factors:	Factors:	Factors:	Factors:	Factors:	Factors:
# of Trials	ABCD	ABCDE	ABCDEF	ABCDEF G	ABCDEF GH	ABCDEF GHJ	ABCDE FGHIK	ABCDEFGH JKL
8	D=ABC	D=AB E=AC	D=AB E=AC F=BC	D=AB E=AC F=BC G=ABC	NA	NA	NA	NA
16	FFD	E=ABCD	E=ABC F=BCD	E=ABC F=BCD G=ACD	E=ABC F=BCD G=ACD H=ABD	E=ABC F=BCD G=ACD H=ABD J=ABCD	E=ABC F=BCD G=ACD H=ABD J=ABCD K=AB	E=ABC F=BCD G=ACD H=ABD J=ABCD K=AB L=AC
32		FFD	F=ABCDE	F=ABCD G=ABDE	F=ABC G=ABD H=BCDE	F=BCDE G=ACDE H=ABDE J=ABCE	F=BCDE G=ACDE H=ABDE J=ABCE K=ABCD	F=ABC G=BCD H=CDE J=ACD K=ADE L=BDE
64			FFD	G=ABCDE F	G=ABCD H=ABEF	G=ABCD H=ACEF J=CDEF	G=BCDF H=ACDF J=ABDE K=ABCE	G=CDE H=ABCD J=ABF K=CDEF L=ADEF

Table 2-3 above is adapted from a similar table provided by Fries and Hunter (p. 601-608). Table 2-3 provides an experimenter with the columns that form a minimum aberration design when the first-order terms in the RS are assigned to the specified columns. The information presented in Table 2-3 does not contain the necessary information to select the best design in a large DOE/RS problem. An extensive literature review failed to uncover information on selecting minimum aberration designs that are larger than those found in Table 2-3.

S. Bisgaard and H.T. Fuller (1995, p. 71 - 74) describe a related method of selecting a design based on defining relations. Their procedure utilizes the confounding structure of a

maximum resolution, minimum aberration design in conjunction with a word processor. Bisgaard and Fuller recommend creating a file which uses integers to refer to the columns of the design and lists the confounding pattern of a general design. Using the *find and replace* function of the word processor, the experimenter can assign the variables sequentially, examining the changing confounding structure as each variable is assigned. While still primarily a manual trial and error method, the use of the word processor eases the task of searching for an acceptable experimental design.

Bisgaard and Fuller's research is a step in the right direction, but the combinatorial nature of the DOE/RS problem suggests investigating heuristic methods.

2.5 Complexity of the DOE/RS Problem

Garey and Johnson (1979) identify the following problem as NP- Complete:

INSTANCE: Graphs $G = (V_1, E_1)$, $H = (V_2, E_2)$.

QUESTION: Does G contain a subgraph isomorphic to H , i.e., a subset

$V \subseteq V_1$ and a subset $E \subseteq E_1$ such that $|V| = |V_2|$ and

$|E| = |E_2|$, and there exists a one-to-one function $f: V_2 \rightarrow V$

satisfying $\{u, v\} \in E_2$, if and only if $\{f(u), f(v)\} \in E$?

A simple graph is a set of nodes or vertices connected by links called edges (or arcs).

In the case of the DOE/RS problem, the RS is analogous to graph H . A Taguchi design graph is analogous to graph G . An isomorphic subgraph is analogous to a subset in set theory: graph H is an isomorphic subgraph of graph G if there exists a subset of the vertices and edges in graph G which correspond to graph H . Therefore solving the DOE/RS problem using a single Taguchi graph is an NP-Complete problem. The optimization methods used in this research were not restricted to a single Taguchi graph. The 0-1 IP formulation, developed in Chapter

Three, searched over all possible fractional factorial designs. Solving the IP formulation is equivalent to solving all possible Taguchi graphs for a given size experiment. Wu and Chen (1992, p. 165) found that 32,768 graphs would be required to identify all the unique experimental designs supported by a 2^{10-5} fractional factorial design. Solving a 2^{10-5} DOE/RS problem using the 0-1 IP formulation would be the equivalent to evaluating 32,768 Taguchi graphs (where evaluating each graph is an NP-Complete problem) and then selecting the best result. Since the DOE/RS problem can be thought of as solving a set of NP-Complete problems, the DOE/RS problem itself is NP-Complete.

The difficulty of solving large NP-Complete problems motivates the development of heuristics in this study. NP-Complete problems grow in complexity so rapidly that computationally intractable problems continue to exist despite improvements in computers and optimization algorithms.

The order complexity is found by estimating the rate at which the number of operations required to solve the problem grows, assuming a worst case search. For combinatorial optimization problems, the worst case search is complete enumeration. In practice, branch-and-bound and other integer programming techniques typically perform more efficiently than complete enumeration, but they cannot *guarantee* better than worst case performance.

For any given DOE/RS problem with K first-order terms and M columns in the design matrix, the number of feasible solutions is equal to the number of unique matchings of K variables to K columns. For the first variable there are M columns to choose from, for the second variable there are M-1, for the third, M-2, and so on until the final variable is assigned to one of M-(K-1) columns. The total number of possible feasible solutions is therefore equal

to $M(M-1)(M-2)\dots(M-K+1)$. Evaluating this expression yields a highest order term M^K . The rate at which the DOE/RS problem grows is therefore on the order of M^K .

Given the limitations of each of the manual methods of optimizing the FFD, it is important to consider the experimenter's alternatives in trying to solve his or her DOE/RS problem.

2.6 Alternatives to Optimizing the Fractional Factorial Designs

If an experimenter wishes to estimate a requirement set using a designed experiment, there are alternatives to optimizing a highly fractionated design. The simplest alternative is to increase the resolution of the experiment by designing a larger experiment. A Resolution V (R_V) design permits the estimation of all two-factor interactions assuming all three-factor and higher terms are negligible. A full factorial design permits estimation of all interaction terms. Unfortunately, these larger experiments require additional experimental resources such as time, money, test objects, and so forth. If the number of test objects is limited, it may not be possible to implement a larger test matrix. Even computer simulation experiments can be expensive if valuable computing assets are occupied for days and weeks longer than is absolutely necessary. If experimental trials are inexpensive and easy, a high-resolution design may be appropriate. Most real world experiments are not inexpensive however, so solving a DOE/RS problem may save thousands of dollars in unnecessary experimental trials.

The experimenter also has the option of using a standard FFD. However, if two or more terms from the requirement set are confounded, then any estimates of the effects of these terms are ambiguous at best. The consequences of failing to unambiguously estimate the effects of each element in the requirement set are clearly dependent on the situation, but uncertainty as

to which factor or interaction term is causing a significant observed effect is never desirable. The risk in applying a design which is not well suited to the requirement set is that the experimenter could very likely fail to extract the maximum amount of information from a given number of experimental trials.

2.7 Research Approach

This dissertation characterizes the performance of several optimization methodologies in solving two-level fractional factorial DOE/RS problems of 16, 32, and 64 trials. DOE/RS problems smaller than 16 trials can be solved using manual methods with reasonable ease. Experiments of 32 and 64 trials, while common, exceed the size for easy use of the current manual methods (Lee 1991, p. 79). Demonstrating that the optimization techniques can effectively construct designs meeting the needs of an experimenter for 64-run experiments establishes that this research provides new capabilities that were previously impractical.

The performance of the solution methodologies is characterized in terms of the computational effort and solution quality (Barr, Golden, Kelly, Resende, and Stewart, 1995). The computational effort characterizes the computer resources expended searching for a solution. The computational effort is measured by counting the number of potential solutions evaluated to solve a particular DOE/RS problem. The computational effort is therefore independent of the speed of the computer. The objective function measures the quality of the solution, with lower values of the objective function preferred over higher values.

What is needed is a means of solving a DOE/RS problem without the limitations of the current manual methods. Chapter Three presents such a method using 0-1 integer programming, while Chapter Four examines heuristic approaches.

III. Optimal Methodology

3.1 Introduction

This chapter develops a formulation that applies a standard 0-1 Integer Program solver to a set of DOE/RS problems and characterizes its performance. An optimality-seeking approach, such as Integer Programming (IP), has the advantage of guaranteeing an optimal solution if the solver runs to completion. Linear Programming (LP) and IP are powerful optimization techniques that have been applied to a wide range of problems. Because there are no non-interactive methodologies for solving any DOE/RS problems, the development of a 0-1 integer programming formulation was an important first step in this analysis. The binary formulation provides the mechanism to utilize an optimum-seeking algorithm in its own right while also providing a baseline to compare the performance of the heuristics developed in this research.

No documented IP formulation is currently available in the literature for solving the DOE/RS problem. Only one IP formulation for any type of DOE problem was documented (Harris, Hoffman, and Yarrow, 1995). The researchers used an IP formulation to solve a minimum-correlation Latin hypercube sampling (MCLHS) problem with two variables.

3.2 Mathematical Model of DOE with Requirement Set (DOE/RS)

In a factorial experiment, some background knowledge may suggest that certain interactions are potentially important. The experimental plan should, therefore, be chosen so that these interactions can be studied without being aliased with each other and with the main effects. The interactions specified by the investigator can be estimated if a design of Resolution V (or higher) is used. Quite often for economic or other reasons, a smaller design is preferred in which some two-factor interactions (2fi's) are aliased with other 2fi's or main

effects. The main question of interest is how to select a fractional factorial design that allows the main effects and the set of interactions specified by the investigator, which are called a *requirement set* (Greenfield 1976), to be estimated. (Wu and Chen 1992, p. 162)

In order to evaluate designs that do not allow estimation of all the terms in the requirement set, the experimenter assigns a value, C_i , to each term of the requirement set. The value reflects the importance of having a clear estimate of the effect of that variable on the process being studied. The Design of Experiments with a Requirement Set (DOE/RS) problem is to select a fractional factorial design that minimizes the sum of the C_i 's associated with confounded terms of the requirement set.

To translate the DOE/RS problem into a mathematical model, an objective function and constraints were developed. The objective function is presented first, followed by the various constraints needed in a 0-1 integer programming formulation. The formulation is structured on a full factorial experimental design of size S . K first-order terms are assigned to columns of the design matrix. The assignment of higher order terms in the RS is mandated by the assignment of the first-order terms.

The terms presented in Table 3-1 are used in developing a mathematical programming formulation of the DOE/RS problem as well as the heuristic methods presented in Chapter Four.

3.2.1 Objective Function. An IP formulation begins with an objective function. In the case of the DOE/RS problem, the objective function is a nonnegative summation function where a solution with a total value of zero indicates that all of the terms of the RS can be estimated.

TABLE 3-1: DOE/RS PROBLEM TERMS

Term	Description
z	z = the objective function value. An objective function value of zero indicates that all the terms in the RS can be estimated and that $C_{\min ab} * y = 0$.
x_i	$x_i = 1$ if the i th term of the RS is confounded with some other term of the RS; otherwise, $x_i = 0$.
$x_{i,j}$	$x_{i,j} = 1$ if the i th element of the RS is assigned to the j th column of the design matrix; otherwise, $x_{i,j} = 0$.
y	$y = 0$ if the design is a specified minimum aberration design, otherwise, $y = 1$.
S	The size of the design selected by the experimenter and therefore the size of the experiment is given by 2^S . $S = K - P$
K	K = the number of experimental variables, and therefore, K = the number of first-order terms in the RS.
P	$(\frac{1}{2})^P$ indicates the fractional size of the design. For example, if $P = 1$ then the design is a half fraction, $P = 2$ indicates a quarter fraction, and so forth.
$C_{\min ab}$	$C_{\min ab}$ is a nonnegative integer weight the experimenter assigns to the importance of selecting a specified minimum aberration design.
C_i	C_i is the positive integer weight that the experimenter assigns to the importance of having a clear estimation of the i th term in the RS relative to other terms in the RS.
W_j	W_j = sum of C_i 's of all interaction terms which contain the variable j .
N	N = the number of terms in the RS.
M	M = the number of columns in the design matrix - 1. The remaining column is reserved to estimate the regression mean. Normally, $M \geq N$; otherwise, it is not possible to clearly estimate each term in the RS.
D	D is a user-specified set of columns in the design matrix. The columns, which comprise D , define a standard maximum resolution, minimum aberration design. When all the first-order terms in RS are assigned to members of D , then the design is a minimum aberration design: this criteria is a sufficient but not necessary condition for a minimum aberration design.

Equation 3-1 is the objective function:

$$\min z = \sum_{i=1}^N C_i x_i + C_{\min ab} y \quad (3-1)$$

The C_i term is a weight assigned to the importance of having a clear estimation of the i th term in the RS, $x_i = 1$ if the i th term of the RS is confounded with some other term of the

RS; otherwise, $x_i = 0$. $C_{\min ab}$ is a weight assigned to the importance of selecting a minimum aberration design; $y = 0$ if the design is a specified minimum aberration design, otherwise, the aberration of the design is uncertain and $y = 1$. N indicates the number of elements in the RS.

The coefficient (C_i) assigned to each element of the RS is proportional to the experimenter's estimate of the relative importance of the element i in the process under study. The largest coefficients are assigned to the most important terms in the RS. One simple scheme for assigning terms would be to assign each interaction term a weight of one and each first-order term a weight of 100. This scheme divides the RS into two groups. All the members of each group are equally important. Any amount of confounding between the interaction terms is preferred over confounding even one additional first-order term. This scheme assumes the number of interaction terms is less than 100.

The weighting scheme can be altered to indicate an experimenter's preference for certain elements within the two groups. If the weights assigned to the terms within a group are different, then the IP will attempt to assign any confounding which does occur to elements with the lowest valued coefficients. This altered approach was used in the study to reduce the number of alternate optimal solutions to the DOE/RS problem. The assumption, which is consistent with Bisgaard and Fuller, is that the terms of the requirement set can be arranged in preferential order. For solutions with the same number and type of confounded terms, this weighting scheme will cause the IP algorithm to attempt to force any confounding to be among the lowest weighted terms. Normally, an experimenter would probably not have such fine distinctions among the groups of terms to be estimated. This set of weights was used to reduce the number of alternate optimal solutions.

An experimenter could use a variety of design analyses and multiattribute criteria theory techniques to develop appropriate “weights” or “costs” (Keeney 1971, Keeney and Raiffa 1976, Stewart 1984, Steuer 1986, Stewart 1987, Barzilai and Golany 1990, Keeney 1992, Clemen 1996).

The existence of an appropriate set of coefficients is guaranteed as long as each term in the objective function is mutually preferentially independent of the other terms (Fishburn 1965, p. 42, Keeney and Raiffa 1976, p. 111, Winston 1994, pp. 772-774). In the case of the DOE/RS problem, the experimenter prefers each of the terms to be clear of confounding regardless of the state of the other terms in the RS. Therefore, the coefficients in each term in the objective function are mutually preferentially independent of the other terms in the objective function. For two designs with the same confounding among the RS, an experimenter has no reason to prefer a nonminimum aberration design. Therefore, the minimum aberration criterion is also mutually preferentially independent of the confounding criteria.

Weights could reflect the cost to the experimenter of changing each variable, a measure of risk associated with each variable, or an estimate of the likelihood of each term being statistically significant. The specific weighting scheme will depend upon the experiment to be conducted. The development of weighting schemes, while clearly important, is not the primary focus of this study. It remains an area for future research which will increase in importance as the approaches outlined here are adopted.

The C_i coefficients enable the search procedures to select designs which contain the confounding structure which best meets the experimenter's design requirements. A design that provides clear estimation of all the terms in the RS is preferred. If such a design cannot be

found, the C_i 's guide the search towards a design which minimizes the sum of the coefficients of the confounded terms, and thus provide a design which allows the experimenter to estimate the "most important" terms in the RS. The coefficients are a proxy for the experimenter's preferences. If the experimenter would prefer the confounding found in design A to that in design B, then the objective function value of design A would be lower than the value of design B. It should be noted that these coefficients are selected to direct the optimization. The amount of specific inferences that can be drawn from the objective function's value is limited to the precision of the estimates of the objective function coefficients. Unless there is reason to believe the C_i 's are tied to specific known values (i.e. actual costs), extreme caution should be used in drawing any conclusions beyond that of a proxy for the experimenter's preferences.

The experimenter also assigns a value to C_{minab} if the experimenter can define *a priori* a set of columns that constitute a minimum aberration design. The C_{minab} term indicates the importance to the experimenter of selecting a maximum resolution, minimum aberration design. Some experimenters might consider maintaining a maximum resolution, minimum aberration design more important than deconflicting some of the less important interaction terms. The magnitude of C_{minab} , relative to the C_i 's, controls the tradeoff between selecting a minimum aberration design and minimizing the confounding of terms. As with the C_i weighting scheme, selection of appropriate C_{minab} values is worthy of additional study.

Aberration is a measure of the confounding which assumes all the interactions of a given order (i.e. second order, third order, etc.) are equally significant. This assumption conflicts with the basis of the DOE/RS problem. Therefore, an experimenter would not normally have a

preference for a specified minimum aberration design and $C_{\min ab}$ would typically be assigned a value of zero.

Selecting a $C_{\min ab}$ with a value greater than the sum of all the C_i 's causes all the designs produced to be one of a set of specified minimum aberration designs. Any subset of the terms in the RS whose C_i 's sum to a value less than $C_{\min ab}$ would potentially be confounded in order to select a minimum aberration design. In contrast, selecting a $C_{\min ab}$ with a value less than the smallest of the C_i 's causes the designs to be selected to satisfy the minimum aberration criteria as a low priority.

Since $C_{\min ab}$ is to be added to the objective function value only when the design selected is not known to meet the maximum resolution, minimum aberration design criteria, another variable, y , is used to flag this condition. Let y equal 0 if the model establishes a design that is known to be a maximum resolution, minimum aberration design. Otherwise, $y = 1$ to indicate that the aberration of the design is uncertain, because the criteria is a sufficient but not necessary condition for the design to be of minimum aberration. If the design is a specified maximum resolution, minimum aberration design then $C_{\min ab} * y = 0$, otherwise, $C_{\min ab} * y = C_{\min ab}$.

By minimizing z , the search procedure attempts to find the design which minimizes the sum of the weights of any aliased terms of the RS, $\sum_{i=1}^N C_i x_i$, while striving to maintain a maximum resolution, minimum aberration design. If every term in the RS can be clearly estimated (no two terms of the RS are confounded), and $y * C_{\min ab} = 0$, the objective function value is zero. This objective is minimized subject to the constraints necessary to express the RS and the structure of the fractional factorial design.

3.2.2 IP Formulation Constraints. The following subsections develop the five types of constraints required to formulate the DOE/RS as a 0-1 IP problem.

3.2.2.1 Assignment Constraints. The first type of constraint ensures that each term in the RS is assigned to exactly one column in the design matrix. The form of the constraint is presented in equation 3-2:

$$\sum_{j=1}^M x_{i,j} = 1 \quad \forall i \in \{1, 2, \dots, N\} \quad (3-2)$$

For each term i in the requirement set, there is one and only one $x_{i,j}$ equal to 1. $x_{i,j} = 1$ implies the i^{th} element of the RS is assigned to the j^{th} column of the design matrix and all other $x_{i,w} = 0$, where $w \neq j$. One such constraint is needed for each term, i , in the RS: N constraints are required.

Constraints of this construction are special ordered sets of type 1 (SOS1). The concept of special ordered sets was pioneered by Beale and Tomlin (1970). While there are 2^M possible combinations of M zero-one variables, there are only M combinations which are feasible for the SOS1. The IP optimization code that was used in this research, CPLEX Ver. 4.3, exploits the structure of special ordered sets to improve the performance of its branch-and-bound algorithm.

3.2.2.2 Resolution III Constraints. A constraint type is needed to prevent two first-order terms from being assigned to the same column of the design matrix. Such a constraint guarantees that the design is at least a Resolution III design. A Resolution III or greater design ensures that the effects from any two independent variables are not confounded with each other. The form of the constraint is presented in equation 3-3:

$$\sum_{i=1}^K x_{i,j} \leq 1 \quad \forall j \in \{1, 2, \dots, M\} \quad (3-3)$$

At most one of the first K terms can be assigned to any column j of the design matrix.

The first K terms are the first-order terms in the RS. This type of constraint is an SOS of Type

2. Rather than considering all 2^K possible combinations for any given column, one need only consider K+1 possible outcomes, because either one of K terms is assigned to the column or no first-order term is assigned to a column. This reduction of the feasible region results in more efficient IP searches. M of this type constraint are required.

3.2.2.3 Minimum Aberration Constraint. If the experimenter wishes to search for a minimum aberration design, a constraint is needed to force the variable y to equal one when the current design is not known to be a specified minimum aberration design. The user specifies a standard set of K columns, which forms a maximum resolution, minimum aberration design when the experimental variables are assigned to these columns.

Box and Draper (1987, pp. 164-165) provide information on such sets of columns for fractional factorial designs, as does Table 2-3. If the first-order terms in the RS are assigned to a user-specified set of K columns (the set D), then the confounding structure is one of a maximum resolution, minimum aberration design. Otherwise, the aberration of the design is unknown and assumed not to be a minimum aberration design. The minimum aberration constraint is sufficient to establish that the design is a minimum aberration design, but is not a necessary condition for a design to be a minimum aberration design. The form of the minimum aberration constraint is given by equation 3-4:

$$Ky + \sum_{x_{i,j} \in D} x_{i,j} \geq K \quad (3-4)$$

The objective function attempts to force $y = 0$ to minimize the term $C_{minab} * y$. y can equal zero only if all the first-order terms in the RS are assigned to the K columns of D , otherwise $y = 1$ and C_{minab} is added to the objective function value. If the sum of the $x_{i,j}$ over the set D is equal to K , then the design is known to be of maximum resolution and minimum aberration. Only one such constraint is required.

If the experimenter wishes to ensure that the design selected has first-order terms assigned to the design columns associated with the factors A, B, C, \dots , then a modified form of this constraint may be used. The user-defined set of columns, D , would contain S columns. Specifically the columns 2^{j-1} for $j = 1, 2, \dots, S$. These are the columns associated with the factors A, B, C , and so on. Appendix H presents the different methods of identifying the columns of the design matrix used in this research.

3.2.2.4 Confounding Constraints. A set of constraints is needed for each i ($i = 1, 2, \dots, N$) element in the RS to force $x_i = 1$ whenever at least one additional element is assigned to the same column in the design matrix. Recall that $x_i = 1$ causes the C_i to be added to the objective function value whenever the i^{th} term is confounded. The formulation for such a constraint is given by equation 3-5:

$$-Nx_i + Nx_{i,j} + \sum_{h=1}^N x_{h,j} \leq N + 1 \quad \forall i \in \{1, 2, \dots, N\}; \forall j \in \{1, 2, \dots, M\} \quad (3-5)$$

Whenever $x_{i,j}$ is equal to zero, the above constraint is not active. In other words, x_i is free to be assigned a value of zero to minimize the objective value. If an $x_{i,j}$ is equal to one, x_i is

forced to equal one if the sum of all the terms in the RS assigned to column j exceeds one. x_i equaling one indicates that the i^{th} term in RS is confounded with at least one other term in the RS. If all the x_i 's are equal to zero, then all the terms in the RS are clear of confounding. In such a case, if either y or $C_{\min ab}$ is zero, then the design is known to be an optimal design for the RS.

Since there are N items in the RS and M design columns, $N \times M$ of these confounding constraints are needed for any DOE/RS formulation. The combinatorial nature of these constraints is a liability to the computational tractability of the proposed model for designs with a large number of key factors.

3.2.2.5 Interaction Constraints. The IP formulation also requires constraints that force the interaction terms in the RS to be assigned to the proper columns. The assignment of the interaction terms is determined by the assignment of the associated first-order terms that comprise each interaction term. For example, assume the two-factor interaction (2fi) term ab is the q^{th} term in the RS. Let a and b be the a^{th} and b^{th} term in the RS. Equation 3-6 forces the ab term to be assigned to the h column when the a is assigned to column j_1 and b is assigned to the column j_2 .

$$x_{a,j_1} + x_{b,j_2} - 2x_{q,h} \leq 1 \quad \forall j_1, j_2 \in \{1, 2, \dots, M\}; \forall q \in \{K+1, K+2, \dots, N\} \quad (3-6)$$

A constraint is needed for each combination of possible assignments of a and b . There are M^2 - M possible combinations for a given 2fi term. Interaction terms containing more than two terms can be accommodated using similar constraints.

The specific constraints were constructed using a vector formulation for the column assignments. Any column assignment is a number less than 2^S . An S-element vector of zeros and ones can represent this number, in base 2 form. For example, if $S = 4$, then $A = (0,0,0,1)$ and $ACD = (1,1,0,1)$, representing the first and thirteenth columns. If first-order terms a and b are assigned to these two columns, then in equation 3-6 $x_{A,1}=1$ and $x_{b,13}=1$. The value of h in equation 3-6 corresponds to the column associated with the interaction term ab . This interaction term can be found by using modulo two addition on each element of the two vectors associated with a and b : $(0,0,0,1) + (1,1,0,1) \bmod 2 = (1,1,0,0)$. This corresponds to the CD column; that is, the twelfth column.

This vector approach was key to writing software that could construct the constraints for all of the interaction terms. The vector approach is a mathematical method for identifying the column associated with the product of two terms that have already been assigned to specific columns. Appendix H contains additional explanation and rationale behind the column numbering approach. The software wrote one constraint for each possible assignment of the first-order terms associated with each interaction term in the requirement set.

If a RS contains only first and second-order terms, the number of constraints required would be $(N - K)(M^2 - M)$. Unfortunately, the combinatorial nature and growth of this constraint type again creates a large constraint set.

3.2.3 Complete IP Formulation. Collectively, the objective function and the constraints for a 2^{K-P} DOE/RS problem with N elements in the RS can be expressed as the following 0-1 IP model:

$$\min z = \sum_{i=1}^N C_i x_i + C_{\min ab} y \quad (3-1)$$

<u>Constraint</u>	<u>No</u> <u>Req</u>	<u>ID</u>
$\sum_{j=1}^M x_{i,j} = 1 \quad \forall i \in \{1, 2, \dots, N\}$	N	(3-2)
$\sum_{i=1}^K x_{i,j} \leq 1 \quad \forall j \in \{1, 2, \dots, M\}$	M	(3-3)
$Ky + \sum_{x_{i,j} \in D} x_{i,j} \geq K$	1	(3-4)
$-Nx_i + Nx_{i,j} + \sum_{h=1}^N x_{h,j} \leq N+1 \quad \forall i \in \{1, 2, \dots, N\}; \forall j \in \{1, 2, \dots, M\}$	NM	(3-5)
$x_{a,j_1} + x_{b,j_2} - 2x_{q,h} \leq 1 \quad \forall j_1, j_2 \in \{1, 2, \dots, M\}; \forall q \in \{K+1, K+2, \dots, N\}$	(N-K)* (M ² -M)	(3-6)
$x_i \in \{0, 1\}, x_{i,j} \in \{0, 1\}, y \in \{0, 1\}$		(3-7)

As stated earlier, if a requirement set contains only first and second-order terms, the number of constraints required would be $N + M + 1 + MN + (N - K)(M^2 - M)$. Even exploiting the computational advantage of the special order sets, this model grows rapidly. For example, a 64-run experiment with 20 variables and 20 interaction terms ($N=40, M=63, K=20$) would require: $[40+63+1+(63)(40)+(20)(63)(62)] = 80,744$ constraints and 2561 binary variables. While the model can be formulated, it rapidly becomes prohibitively large. In

this example there are 2^{561} (approximately 10^{770}) possible solutions. Not all of these solutions are feasible. The total number of feasible solutions for this DOE/RS problem is equal to the number of ways to assign each of the 20 first-order terms in the RS to its own column (out of the 63 columns) of the experimental design. The first term can be assigned to one of 63 columns, the second to one of 62 columns and so forth. The total number of feasible solutions is $63 \times 62 \times 61 \times \dots \times 44$ which is equal to $63!/43!$ or about 3×10^{34} . While this reduction is quite significant when compared to 10^{770} , it still represents a very large feasible region to be examined.

In order to solve large DOE/RS problems as zero-one IPs, a matrix generator was written which generates an input file for CPLEX in CPLEX's "LP format". The LP format allows CPLEX to read problems written in algebraic form as they typically appear in texts. The interaction constraints are unique to the DOE/RS problem and required specialized code to write a set of constraints based on the confounding of the fractional factorial design. One general program was written as part of this research that creates an appropriate IP formulation for the specified DOE/RS problem. The program requires inputs that identify the RS and the size of the experimental design. This program (with embedded inputs such as the RS) is listed and documented in Appendix D. An example of the output associated with the IP is presented in Appendix C.

3.2.4 Example Problem. This section presents a small notional DOE/RS problem to illustrate the IP formulation. While the proposed approach would not be used on a problem of this size, the problem is useful to illustrate the model. The requirement set for this problem consists of two variables A and B and their interaction term AB. In this example A is x_1 , B is x_2 , and the

interaction term is x_3 . The experimental design is a 2^{2-0} design. There are three columns in this experimental design:

TABLE 3-2: 2^{2-0} EXPERIMENTAL DESIGN

Experimental Run	Column 1 (C1)	Column 2 (C2)	Column 3 (C3)
1	+1	+1	+1
2	-1	+1	-1
3	+1	-1	-1
4	-1	-1	+1

Each of the terms (x_1, x_2, x_3) must be assigned to a column. In order to demonstrate the minimum aberration constraint, assume that the set of columns {C1, C2} has been identified as a minimum aberration design. The column order for this design is $A = C1, B = C2$, and $AB = C3$.

3.2.4.1 Objective Function. Minimize:

$$\text{Obj: } 100 x_1 + 100 x_2 + 20 x_3 + 10 y \quad (3-8)$$

These weights imply that obtaining clear estimates of both A and B are equally important. Furthermore, unconfounded estimates of the first-order terms are more important than clearly estimating the AB interaction term or developing a minimum aberration design. The experimenter also prefers a clear estimate of the interaction term over satisfying the minimum aberration constraint. Finally, the positive weight for y indicates that the experimenter does have a preference for a minimum aberration design; otherwise, this weight would be zero.

We will now consider the constraints:

3.2.4.2 Assignment Constraints.

$$x_{1,1} + x_{1,2} + x_{1,3} = 1 \quad (3-9)$$

$$x_{2,1} + x_{2,2} + x_{2,3} = 1 \quad (3-10)$$

$$x_{3,1} + x_{3,2} + x_{3,3} = 1 \quad (3-11)$$

Equations 3-9 through 3-11 are the assignment constraints. $x_{3,1} = 1$ if x_3 is assigned to column

1. Likewise, $x_{3,2} = 1$ if x_3 is assigned to column 2 and $x_{3,3} = 1$ if x_3 is assigned to column 3.

Equation 3-11 forces x_3 to be assigned to one and only one column. Equation 3-9 does the same thing for x_1 and equation 3-10 does the same for x_2 .

3.2.4.3 Resolution III Constraints.

$$x_{1,1} + x_{2,1} \leq 1 \quad (3-12)$$

$$x_{1,2} + x_{2,2} \leq 1 \quad (3-13)$$

$$x_{1,3} + x_{2,3} \leq 1 \quad (3-14)$$

Equations 3-12 through 3-14 are the Resolution III constraints. In order to ensure that the design is at least Resolution III, two or more first-order terms must never be assigned to the same column. In this IP formulation, equation 3-12 prevents both x_1 and x_2 from being assigned to column 1. If they were both assigned to column 1, then $x_{1,1}$ and $x_{2,1}$ would be one and the sum would be greater than one, which violates equation 3-12. Equation 3-12 allows either x_1 , x_2 , or neither of the two to be assigned to column one. Likewise, equations 3-13 and 3-14 ensure that both x_1 and x_2 are not assigned to column 2 and column 3 respectively.

3.2.4.4 Confounding Constraints.

$$-3x_1 + 3x_{1,1} + x_{1,1} + x_{2,1} + x_{3,1} \leq 4 \quad (3-15)$$

$$-3x_1 + 3x_{1,2} + x_{1,2} + x_{2,2} + x_{3,2} \leq 4 \quad (3-16)$$

$$-3x_1 + 3x_{1,3} + x_{1,3} + x_{2,3} + x_{3,3} \leq 4 \quad (3-17)$$

$$-3x_2 + 3x_{2,1} + x_{1,1} + x_{2,1} + x_{3,1} \leq 4 \quad (3-18)$$

$$-3x_2 + 3x_{2,2} + x_{1,2} + x_{2,2} + x_{3,2} \leq 4 \quad (3-19)$$

$$-3x_2 + 3x_{2,3} + x_{1,3} + x_{2,3} + x_{3,3} \leq 4 \quad (3-20)$$

$$-3x_3 + 3x_{3,1} + x_{1,1} + x_{2,1} + x_{3,1} \leq 4 \quad (3-21)$$

$$-3x_3 + 3x_{3,2} + x_{1,2} + x_{2,2} + x_{3,2} \leq 4 \quad (3-22)$$

$$-3x_3 + 3x_{3,3} + x_{1,3} + x_{2,3} + x_{3,3} \leq 4 \quad (3-23)$$

Equations 3-15 through 3-23 are the confounding constraints. Their function is to force x_i to equal 1 if some other term in the requirement set is assigned to the same column x_i is assigned to. For example equation 3-15 forces $x_1 = 1$ if x_1 is assigned to column 1 (i.e. $x_{1,1} = 1$) and some other term in the RS is also assigned to column 1 (i.e. $x_{1,1} + x_{2,1} + x_{3,1} \geq 2$). If x_1 is the only term assigned to column one then $x_{1,1} + x_{1,1} + x_{2,1} + x_{3,1} = 4$ and x_1 is not constrained. The objective function will try to set x_1 to zero unless a constraint forces it to one. If $x_{1,1} = 0$ and all other $x_{i,1}$ terms are zero, then x_1 will be zero. The A term is not confounded with any other term in the RS. If x_1 is not assigned to column 1 ($x_{1,1} = 0$) then x_1 is not constrained. If none of the constraints force $x_1 = 1$, then the objective function will set $x_1 = 0$ in order to minimize the objective function value.

Equations 3-16 and 3-17 cover the cases where x_1 is assigned to columns 2 and 3, respectively. Equations 3-18, 3-19, and 3-20 deal with x_2 and columns 1, 2, and 3. Equations 3-21, 3-22, and 3-23 cover the cases where x_3 is assigned to columns 1, 2, or 3.

3.2.4.5 Interaction Constraints.

$$x_{1,1} + x_{2,2} - 2 x_{3,3} \leq 1 \quad (3-24)$$

$$x_{1,1} + x_{2,3} - 2 x_{3,2} \leq 1 \quad (3-25)$$

$$x_{1,2} + x_{2,1} - 2 x_{3,3} \leq 1 \quad (3-26)$$

$$x_{1,2} + x_{2,3} - 2 x_{3,1} \leq 1 \quad (3-27)$$

$$x_{1,3} + x_{2,1} - 2 x_{3,2} \leq 1 \quad (3-28)$$

$$x_{1,3} + x_{2,2} - 2 x_{3,1} \leq 1 \quad (3-29)$$

Equations 3-24 through 3-29 are the interaction constraints. Their function is to force the interaction term x_3 (AB) to be assigned to the appropriate column. The appropriate column is a function of the column assignments for x_1 (A) and x_2 (B). Equation 3-24 can be interpreted as: "If x_1 is assigned to column 1 (i.e. $x_{1,1} = 1$) and x_2 is assigned to column 2 (i.e. $x_{2,2} = 1$),

then x_3 must be assigned to column 3 (i.e. $x_{3,3} = 1$), otherwise $x_{3,3}$ is unconstrained." There are $M*(M-1)$ possible assignments for x_1 and x_2 . In this design, M is three and so six of these constraints are required for each interaction term in the RS. In a design where M is 63, then each interaction term requires $63*62$ or 3,906 of this type of constraint. If there are 20 interaction terms in the RS, then the number of interaction constraints would be 78,012. It is this combinatorial increase in the number of constraints that makes the IP formulation of large DOE/RS problems challenging to solve.

3.2.4.6 Minimum Aberration Constraint.

$$2y + x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2} \geq 2 \quad (3-30)$$

Equation 3-30 is the minimum aberration constraint. It is trivial in this problem but in larger problems it could be of value to the experimenter. In a 2^{4-1} design for example, columns 1, 2, 4, and 7 form the set of columns associated with a known minimum aberration design. As long as each of the four first-order terms in the RS are assigned to one of these four columns, then the design will be a minimum aberration design, otherwise, the aberration of the design is unknown; that is, the design may or may not be a minimum aberration design.

3.2.4.7 Integer Constraints. All the variables are 0-1 integers.

3.2.4.8 Optimal Solution. One solution to this example problem is $x_{1,1} = 1$, $x_{2,2} = 1$, $x_{3,3} = 1$. All other variables are zero and the objective function has a value of zero. In this solution, the variable A is assigned to the first column, B is assigned to the second column and AB is associated with column 3. An alternate optimal solution, for example, is $x_{1,2} = 1$, $x_{2,1} = 1$, and $x_{3,3} = 1$ and all other variables equal to zero. There are six different optimal solutions to this problem.

3.3 The IP Experiment

The experiment evaluated the IP formulation over a range of DOE/RS problems that are difficult and cumbersome to solve with current methods. The level of difficulty is a function of the size of the fractional factorial design and the number of terms in the RS. Therefore, this research focuses on FFD's of 16, 32, and 64 runs. These designs are commonly used and are sufficiently large to be difficult using manual methods. Solving problems of this size is useful in its own right and establishes that computer-based methods could be used to solve some of these large DOE/RS problems.

It should be noted that the more terms in the RS, the more difficult the DOE/RS problem. The factor S is a measure of the size of the experimental design (i.e. the number of runs is equal to 2^S). The factor S is therefore a measure of the size of the solution space that must be searched. The size of the space is likely to have an impact on the performance of the solution methodologies.

A *saturated design* is one in which the number of columns in the design matrix is equal to the number of terms to be evaluated. For example, an eight-run experiment with a RS of (a, b, c, d, e, ab, cd) is fully saturated since only seven terms (plus the mean) can be estimated with eight data points. This research focuses on designs that are saturated or nearly saturated in order to examine the most challenging DOE/RS problems given the specified S . A DOE/RS problem that is less saturated than a comparable problem is easier to solve.

The DOE/RS problems used were selected to be among the most difficult for their class. For any fixed design size, removing an element from the RS must make the problem easier (or at least no more difficult) to solve. In the eight-run experiment mentioned above,

removing the cd interaction term from the RS makes the problem easy to solve. The new DOE/RS problem can be solved by assigning $a=A$, $b=B$, $c=C$, $ab=AB$, and d and e can be assigned to any two of the remaining three columns. Removing an element from a RS eliminates constraints from an integer program formulation. %UC is the percentage of unassigned columns which is equal to $(2^S - 1 - N) / (2^S - 1)$. For this research, the percentage of unassigned columns (%UC) ranges from zero (a fully saturated design) to 20%.

TABLE 3-3: EXPERIMENTAL FACTORS

Factor	Low Level	Medium Level	High Level
S	4	5	6
%UC	0%	10%	20%

S and %UC were evaluated at the three different levels depicted in Table 3-3. Thus, nine baseline DOE/RS experiments were established. Specific RS terms were developed to have all experimental variables included in the set of interaction terms. The number of experimental variables was chosen as 7, 12, and 17 for $S = 4, 5$, and 6, respectively.

The output variables characterize the performance of the solution methodologies. Performance has two principal qualities. The first quality is simply how near the design is to providing the maximum possible number of clear estimates for all the terms in the RS. The metric "percentage of relative reduction" is calculated as shown in equation 3-31. The variables with the "*" are variable values associated with the best known solution; the variables with the "/" are the variable values associated with the solution the metric is measuring. This metric is a useful approximate measure for the quality of the solution; however, this measure is sensitive to the magnitude of the weights involved. The metric will tend towards 100% as the weights of the unconfounded terms increase relative to the weights of the confounded terms.

$$1 - \frac{\left(\sum_{i=1}^N C_i (1-x_i^*) + C_{\minab} (1-y^*) \right) - \left(\sum_{i=1}^N C_i (1-x_i') + C_{\minab} (1-y') \right)}{\left(\sum_{i=1}^N C_i (1-x_i^*) + C_{\minab} (1-y^*) \right)} \quad (3-31)$$

Other metrics that were not captured in the current data but might be useful in future research include the number of first and second-order terms confounded. No approved methodology was found that indicated how to measure the “quality” of a confounded design.

The second consideration relates to the computational effort spent to obtain the design.

Table 3-4 summarizes the dependent variables for this research.

TABLE 3-4: DEPENDENT VARIABLES

Variable
CPU time
Functional evaluations
Objective Function Value
of variables and constraints

3.3.1 The Baseline Experimental Design. Based on the independent variables described earlier, Table 3-5 provides the baseline designs with their RS's. These designs offer a sample of the saturated and nearly saturated DOE/RS problems. By evaluating the performance of a solution methodology against these representative DOE/RS problems, this research estimated the performance of the IP (or other solution methodology) in solving DOE/RS problems.

The designs are also referred to by identifying the number of columns in the design matrix and the number of terms in the RS. DOE/RS problems one through nine can therefore also be identified as: 16/12, 16/13, 16/15, 32/25, 32/28, 32/31, 64/51, 64/57, 64/63. Notice that an estimate for the mean is not included in the RS. A fully saturated DOE/RS problem has a RS that contains one less term than the number of columns in the design matrix (16/15, 32/31 and 64/63 in the experiment).

The weights in the IP formulation indicate the experimenter's preference for having a clear estimate of each term in the RS. The term with the largest weight is the most important term. If two terms have the same weight, then the experimenter is indifferent to which of them is confounded if one of the two must be confounded.

TABLE 3-5: THE BASELINE EXPERIMENTAL DESIGN

Problem	S	%UC	ID	RS : term (cost)
1	4	20%	16/12	a, b ,c, d, e, f, g, ab, cd, ef, ag, bg
2	4	10%	16/13	a, b ,c, d, e, f, g, ab, cd, ef, ag, bg, cf
3	4	0%	16/15	a, b ,c, d, e, f, g, ab, cd, ef, ag, bg, cf, de, ac
4	5	20%	32/25	a, b ,c, d, e, f, g, h, i, j, k, l, ab, cd, ef, gh, ij, kl, al, bk, cj, di, eh, fg, bg,
5	5	10%	32/28	a, b ,c, d, e, f, g, h, i, j, k, l, ab, cd, ef, gh, ij, kl, al, bk, cj, di, eh, fg, bg, ad, be, cf
6	5	0%	32/31	a, b ,c, d, e, f, g, h, i, j, k, l, ab, cd, ef, gh, ij, kl, al, bk, cj, di, eh, fg, bg, ad, be, cf, dg, ae, bf
7	6	20%	64/51	a, b ,c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, ab, cd, ef, gh, ij, kl, mn, op, aq, ac, bd, ce, df, eg, fh, gi, hj, ik, jl, km, ln, mo, np, oq, ad, be, cf, dg, eh, fi, gj, hk, il, jm
8	6	10%	64/57	a, b ,c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, ab, cd, ef, gh, ij, kl, mn, op, aq, ac, bd, ce, df, eg, fh, gi, hj, ik, jl, km, ln, mo, np, oq, ad, be, cf, dg, eh, fi, gj, hk, il, jm, kn, lo, mp, nq, ae, bf
9	6	0%	64/63	a, b ,c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, ab, cd, ef, gh, ij, kl, mn, op, aq, ac, bd, ce, df, eg, fh, gi, hj, ik, jl, km, ln, mo, np, oq, ad, be, cf, dg, eh, fi, gj, hk, il, jm, kn, lo, mp, nq, ae, bf, cg, dh, ei, fj, gk, hl

The nominal weight associated with each term I of the RS is i . For first-order terms, the corresponding weight for each term I is $i + 100$. In the first experiment a is assigned a value of 101, b is assigned 102, and so forth until the twelfth and final term, bg, is assigned a value of 12.

3.3.2 IP Results. The 0-1 IP formulation correctly solved the DOE/RS problem presented by Bisgaard and Fuller (1995, p. 72) and described in detail in Appendix C. The Bisgaard and Fuller problem had eight experimental factors and three interaction terms in the RS

(indicated as experimental design 16/11). The IP formulation required 177 variables and 822 constraints. Modifying the code to produce the IP formulation and then running the IP formulation took less than thirty minutes. All the terms in the RS were clearly estimated. The IP solution technique had excellent results solving the larger, more complex, problems examined in this study as long as the capacity of the computer (usually in terms of RAM) was sufficient. Problems six and nine were the most difficult to solve using the IP method. The branch-and-bound search exhausted the available computer resources before the search completed. Whenever the 0-1 IP failed to establish optimality, the best incumbent integer solution was recorded as the IP result.

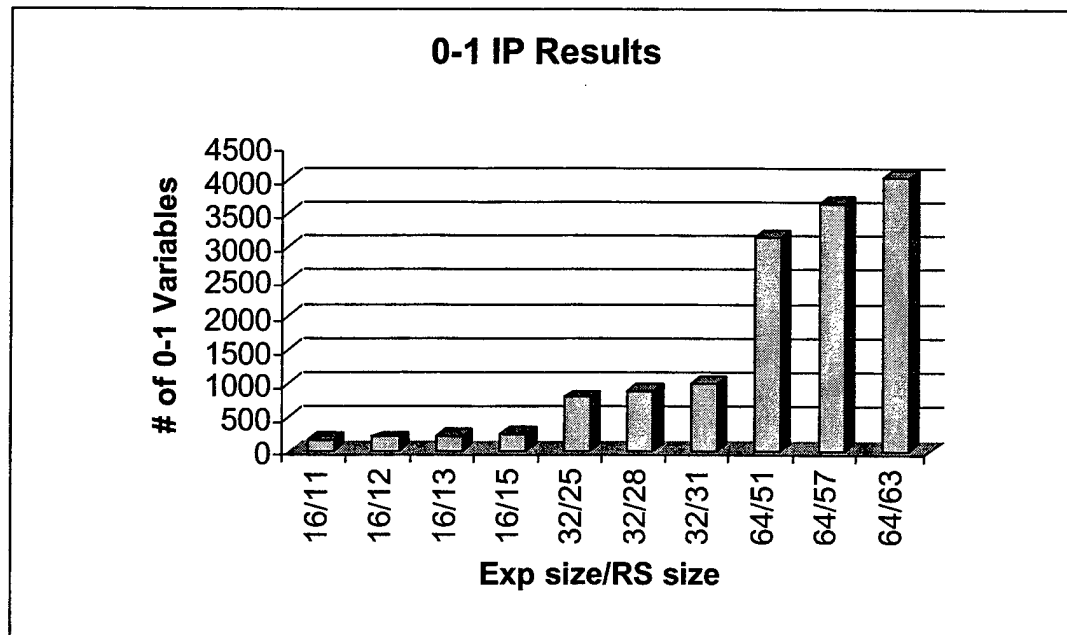


Figure 3-1. Number of Variables vs. Experiment

Figures 3-1 and 3-2 relate the number of variables and constraints for the nine baseline experiments. As expected, the number of variables and constraints increased dramatically as the size of the DOE/RS problem increased. The number of variables increases by a factor of

four each time the size of the DOE/RS problem doubles. The number of constraints increases by an order of magnitude each time the problem size doubles. The number of constraints and variables for the three largest IP formulations was sufficiently large to overwhelm the memory capacity of the SPARC 10 with 64 megabytes of RAM. Memory limitations were also experienced when using a SPARC Ultra with 256 megabytes of RAM.

Table 3-6 presents the results of the IP methodology as applied to the DOE/RS problem. The objective function value is indicated in the "Cost Function" column. The "% of Relative Reduction" column compares the IP solution value to the value of the best solution found using all the methodologies, as a percentage (see equation 3-31). Averaging over all nine

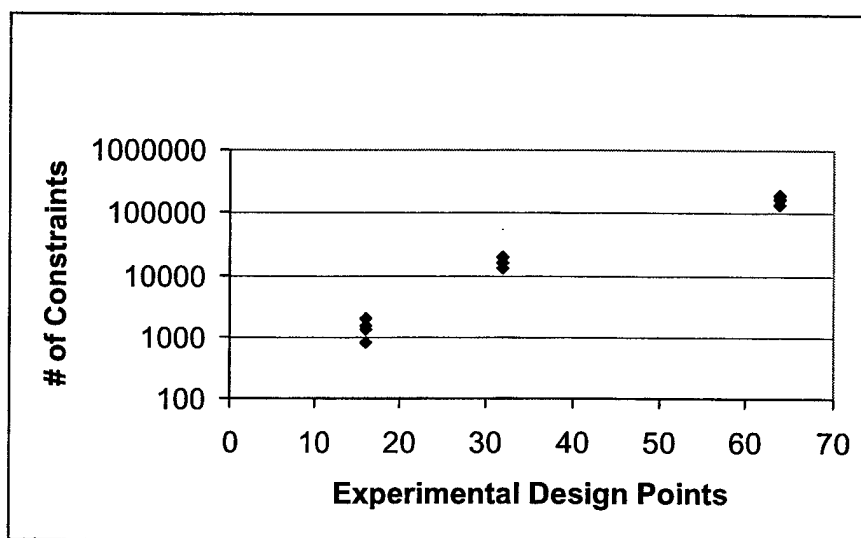


Figure 3-2. Number of Constraints vs. Number of Experimental Design Points

problems, the IP solutions were within 98.45% of the best known value. In the three cases where the IP methodology did not prove optimality, the best solution found by all the methods was used as a surrogate for the value of the optimal solution. The IP value can be less than

100% only if the IP search was terminated due to computer constraints before the search was completed and the heuristic approaches in Chapter Four found a better solution. The last column indicates if the search was completed.

As Table 3-6 shows, not all of the problems were solved to completion. The IP branch-and-bound method terminated the search only when a zero-valued solution was found. The search could terminate because the objective function is nonnegative and therefore a feasible solution with a value of zero must be an optimal solution. The IP methodology failed to terminate when trying to solve the three saturated DOE/RS problems: 16/15, 32/31, and 64/63. For those problems where the search terminated due to limitations on the number of active nodes in the search tree (computer limitations), the solution values presented are the best integer solution found by the IP search before the search was terminated.

The IP problems were run on SPARC 10 and SPARC Ultra workstations. These results required up to two weeks to run. The runs were terminated when the branch-and-bound search reached a limit in the number of functional evaluations or the number of active nodes in the search tree.

The availability of a powerful processor and a mathematical programming code such as CPLEX that can take advantage of special ordered sets are factors in favor of using the 0-1 IP approach. If the experiment was sufficiently costly and important, the use of a more powerful computer and computational time to solve the DOE/RS might well be justified to prove optimality.

TABLE 3-6: THE RESULTS OF THE IP METHODOLOGY

Ex/RS	Objective Function (OF)	% of Relative Reduction	Functional evaluations	Completed (Y/N)
16/11	0	100%	14,384	Yes
16/12	17	100%	1,620,000	Yes
16/13	17	100%	1,670,000	Yes
16/15	51	99%	6,890,000	No (Node Limit)
32/25	0	100%	23,576	Yes
32/28	0	100%	1,120,000	Yes
32/31	198	88%	3,000,000	No (Evaluation Limit)
64/51	0	100%	46,290	Yes
64/57	0	100%	2,300,000	Yes
64/63	2000	99%	500,000	No (Node Limit)

Parallel processing might be applied to increase the computational resources available to solve the DOE/RS problem. In addition, the IP method could use as an initial solution the best solution found by the heuristic methods to be discussed in the next chapter. The IP formulation is the only automated method in this research that could prove a non-zero valued solution is an optimal solution.

3.3.2.1 Specific Solutions. The specific solutions to two IP problems are presented in this section. Both the IP variable values and a DOE column assignment are presented for each solution. Table 3-1 and Appendix H contain reference information on the nomenclature.

3.3.2.1.1 Solution to Problem One. The RS for Problem One (16/12) is {a, b, c, d, e, f, g, ab, cd, ef, ag, bg}. The IP solution is: $1 = x_8 = x_9 = x_{1,4} = x_{2,8} = x_{3,2} = x_{4,14} = x_{5,1} = x_{6,10} = x_{7,7}$

$= x_{8,12} = x_{9,12} = x_{10,11} = x_{11,3} = x_{12,15}$. All other variables have a value of zero. The objective

function value is 17. Notice that $x_8=1$ and $x_9=1$ indicates that ab and cd are confounded.

TABLE 3-7: IP SOLUTION TO PROBLEM ONE(16/12)

RS Term	Column # Assigned	Vector Representation	Letter Representation	Term is Confounded (Y/N)
A	4 ($x_{1,4}=1$)	(0,1,0,0)	C	No
B	8 ($x_{2,8}=1$)	(1,0,0,0)	D	No
C	2 ($x_{3,2}=1$)	(0,0,1,0)	B	No
D	14 ($x_{4,14}=1$)	(1,1,1,0)	BCD	No
E	1 ($x_{5,1}=1$)	(0,0,0,1)	A	No
F	10 ($x_{6,10}=1$)	(1,0,1,0)	BD	No
G	7 ($x_{7,7}=1$)	(0,1,1,1)	ABC	No
Ab	12 ($x_{8,12}=1$)	(1,1,0,0)	CD	Yes ($x_8 = 1$)
Cd	12 ($x_{9,12}=1$)	(1,1,0,0)	CD	Yes ($x_9 = 1$)
Ef	11 ($x_{10,11}=1$)	(1,0,1,1)	ABD	No
Ag	3 ($x_{11,3}=1$)	(0,0,1,1)	AB	No
Bg	15 ($x_{12,15}=1$)	(1,1,1,1)	ABCD	No

Appendix H presents a complete mapping of the columns identified in the IP solution to columns in a full factorial design of 16, 32, and 64 trials. In Problem One, $= x_{1,4} = 1$ implies that $a \rightarrow C$. Table 3-7 summarizes the solution to Problem One.

Problem One was run a second time with an additional constraint which forced a standard design with four first-order terms to be assigned to columns A, B, C, D. This

formulation solved to completion proving that there is no standard two-level fractional factorial design of 16 runs which allows the clear estimation of all the terms in the RS.

3.3.2.1.2 Solution to Problem Two. Problem Two (16/13) is similar to Problem One except for the addition of the cf term in the RS: the RS is {a, b, c, d, e, f, g, ab, cd, ef, ag, bg, cf}. The IP solution is: $1 = x_8 = x_9 = x_{1,8} = x_{2,14} = x_{3,2} = x_{4,4} = x_{5,12} = x_{6,7} = x_{7,1} = x_{8,6} = x_{9,6} = x_{10,11} = x_{11,9} = x_{12,15} = x_{13,5}$. All other variables have a value of zero. The objective function value is 17. Once again, ab and cd are confounded.

The DOE variable to column assignment is as follows: $a \rightarrow D$, $b \rightarrow BCD$, $c \rightarrow B$, $d \rightarrow C$, $e \rightarrow CD$, $f \rightarrow ABC$, $g \rightarrow A$. This assignment of the first-order terms of the RS implies the following column assignments for the interaction terms: $ab \rightarrow BC$, $cd \rightarrow BC$, $ef \rightarrow ABD$, $ag \rightarrow AD$, $bg \rightarrow ABCD$, $cf \rightarrow ABD$. The solution to Problem Two is an alternative, equally good, solution to Problem One.

3.4 Summary

Overall, this research has shown the IP methodology to be an effective method for solving the DOE/RS problem. The disadvantage is that the computational effort to solve large DOE/RS problems may exceed the available resources in terms of computer time and memory. Specialized code was developed to write the IP formulation and a general-purpose IP solver was used to solve the DOE/RS formulation. The heuristics presented in the following chapter require fewer computational resources but the solutions (other than a zero-valued solution) are not guaranteed to be optimal.

Now that a formulation has been established, future research can focus on developing special purpose optimal solvers for this class of problem. In addition, ever increasing computer

capabilities will allow larger designs to be considered with commercial, off-the-shelf, integer programming solvers. The IP solved to optimality all but the saturated problems. DOE/RS problems 32/35, 64/51, and 16/11 were solved quickly and efficiently. Increasing computing speeds and memory capacity will allow optimal solutions for larger and saturated problems.

IV. Heuristic Methodologies

4.1 Introduction

The research presented in Chapter Three identified a set of DOE/RS problems with computational requirements that can rapidly overwhelm a general IP solver. NP-Complete problems, including the DOE/RS problem, often become intractable to optimization algorithms as they become large. Heuristics are used when an algorithm is unsuitable because of timeliness requirements, computational limitations, or ill-defined problem structures. This chapter presents background on heuristics and then develops three heuristic methodologies for solving the DOE/RS problem. Column Assignment Via Examination (CAVE) [a greedy heuristic], and Iterative Column Assignment Via Examination (ICAVE) [an extension of the CAVE heuristic], are designed to utilize the structure of the experimental design to search for a good solution. A general-purpose heuristic is then presented for comparison: Design of Experiments using Simulated Anneling (DESA) [based on the simulated annealing heuristic approach].

All of the heuristics presented in this chapter were coded using compiled QuickBASIC. The compiled code was run on a variety of 486 and Pentium 100 PC's. The source code for each of the heuristics is presented in the Appendices.

4.2 Heuristics

The word "Heuristic" is derived from the Greek word *heuriskein* meaning to discover (Daellenbach, George, and McNickle 1983, p. 647). A heuristic is a well-defined set of steps for quickly identifying a high-quality solution for a given problem (Barr, Golden, Kelly, Resende, and Stewart, 1995, p. 3).

Sangiovanni-Vincentelli (1991, p. 295) separates heuristic methods into two classes: a class that computes a solution constructively starting from raw data (greedy heuristics), and a class that iteratively improves upon an existing solution (iterative improvement heuristics). This research considers both types of heuristics to solve the DOE/RS problem.

4.2.1 Evaluation of Heuristics. Barr, Golden, Kelly, Resende, and Stewart (1995, p. 7-9) provide a comprehensive set of guidelines for evaluating and reporting on heuristics. The effectiveness of any proposed methodology for solving a given class of problems can be demonstrated by theoretical analysis and empirical testing.

Although no set standards are available for evaluating algorithmic research, advances are generally attributable to:

- **Speed** - producing high-quality solutions quicker than other approaches;
- **Accuracy** - identifying higher-quality solutions than other approaches;
- **Simplicity** - easy to implement;
- **Innovation** - new and creative in its own right

One of the most popular and illuminating exhibits of heuristic performance is a graph of solution quality as a function of time (Barr, Golden, Kelly, Resende, and Stewart 1995, p. 9). Such a graph highlights two important performance measures for a heuristic: the quality of the solutions found, and how the quality of the solutions changes as a function of computational effort. To be independent of the platform on which a heuristic is run, computational effort is often reported in iterations of the algorithm or number of functional evaluations (Barr, Golden, Kelly, Resende, and Stewart 1995, p. 11). This report identifies the number of functional evaluations as the metric to compare with different solution algorithms.

4.3 CAVE / ICAVE Heuristics

The development of the Column Assignment Via Examination (CAVE) heuristic was a key contribution of this research. It exploits the underlying structure of the fractional factorial design in order to construct a solution to a DOE/RS problem. The structure of a 2^{K-P} fractional factorial design is a full factorial design for S terms, where $S = K - P$. CAVE utilizes the structure of the full factorial design to attempt to construct a design which minimizes the sum of the cost coefficients associated with the confounded terms in the RS, by sequentially assigning experimental variables to columns in the design matrix. For ease of discussion, we will equate each first-order term in the RS to an experimental variable. The term “the variables” then refers to the set of all first-order terms in the RS.

A vector with S elements can represent the column assignment for each first-order term. The notation is the same as presented in section 3.2.2.5. Each element is either a zero or a one. If each column in the design is identified by a numerical value: $A=1$, $B=2$, $AB=3$, $C=4$ and so forth, then the numerical value of the column associated with any S -vector representation can be determined by treating the vector as a base 2 numeral. The column associated with $(1,0,1,0)$ is calculated by adding $1*8 + 0*4 + 1*2 + 0*1$ and this column is therefore the 10th column of the design. This column is also known as the BD column, as depicted in Appendix H.

In a 16-run experiment, S is equal to 4. If the fifth element (a first-order term) in the RS is assigned to the ABD column, $(1,0,1,1)$ represents the ABD column. Once all the first-order terms are assigned, the higher-order terms are assigned by using mod 2-vector addition. For example, the correct column assignment of a 2fi where the first-order terms have been assigned

columns (1,1,0,0) and (1,0,0,1) is : $(1+1=0, 1+0=1, 0+0=0, 0+1=1)$ or (0,1,0,1) or the AC column. $1+1 \bmod 2 = 0$ because in mod 2 addition any even sum is equal to zero and any odd sum is equal to one.

```

Procedure: CAVE
begin
(1) Calculate  $W_j$ 's
(2) Rank variables by  $W_i$  in descending order
(3) Assign S variables by rank order
(4) Assign interaction terms
Do until all variables assigned
(5) Assign next variable by rank order
(6) Assign interaction terms
End Do
(7) Print solution
end;

```

Figure 4-1. Pseudocode for CAVE Heuristic

4.3.1 CAVE. The CAVE procedure is presented in Figure 4-1 in pseudocode. The CAVE procedure solves the DOE/RS problem that is a bipartite matching problem with constraints. The C_i 's are the positive integer weights that the experimenter assigns to the importance of having a clear estimation of the i th term in the RS. W_j is the sum of C_i 's of all interaction terms that contain the variable j .

The variable with the largest W_j is ranked highest and is always assigned to the column associated with the factor A. The second highest ranked variable is assigned to the column associated with factor B, and so forth. Ties between ranked variables are broken via random uniform selection.

After S highest ranked variables are assigned, any interaction terms composed exclusively of these S experimental variables are assigned to corresponding interaction columns

in the design matrix. Since the design matrix contains a unique column for all possible interaction terms of the S factors, each of the S variables and all their interaction terms will be assigned unique columns.

Each remaining variable is assigned to the column that causes the smallest increase in the objective function. Once all variables have been assigned to columns, a design is complete. The objective function is the same as equation 3-1 presented in Chapter Three:

$$\min z = \sum_{i=1}^N C_i x_i + C_{\min ab} y \quad (4-1)$$

The code implementing CAVE is presented in Appendix E.

While CAVE is a straightforward approach, the following Theorem of CAVE Optimality Conditions serves two purposes. First, it proves the conditions under which CAVE is guaranteed to produce an optimal solution. Such information is important, if such a condition exists, for any new heuristic. Second, it provides the foundation for why CAVE can be expected to produce good solutions even when optimality cannot be guaranteed.

Theorem of CAVE Optimality Conditions: Given a RS such that the set of all the interaction terms in the RS contains no more than S first-order terms, and the ordinality of the RS does not exceed $2^S - 1$, then the CAVE procedure creates a design which allows clear estimation of every term in the RS.

Proof by Construction: Let K be the number of first-order terms in the RS. Let P be an integer such that $S = K - P$.

CAVE assigns the S variables associated with the interaction terms in the RS to the first S main effects columns (e.g. A, B, C, ...).

The interaction terms for these S variables (therefore all the interaction terms in the RS) are assigned by CAVE to their appropriate column. Since the design is a full factorial design for S variables, each possible interaction term has its own unique column.

Each of the remaining P variables will be assigned to one of the remaining columns of the design. Since these terms are first-order terms and do not appear in the interaction terms, they can be arbitrarily assigned to any unassigned column without changing the confounding nature of the design.

At this point all of the terms in the RS have been assigned to unique columns and no two terms are confounded with each other. Therefore, the CAVE procedure created a design that allows clear estimation of every term in the RS. QED

4.3.1.1 Solving an Example Problem Using CAVE. The CAVE procedure is applied to the Bisgaard & Fuller problem described in Appendix C. The RS is {A, B, C, D, E, F, G, H, AB, AD, BD}, with corresponding weights of {100, 100, 100, 100, 100, 100, 100, 100, 10, 9, 8}. The weighting is consistent with the literature in that it expresses a preference among the three interaction terms but treats all the first-order terms as equally important (Bisgaard and Fuller, 1995). A 2^{8-4} experimental design has eight factors and 16 runs. As shown in Table 2-3 the columns for an eight-factor, 16-run minimum aberration design are A, B, C, D, ABC, BCD, ACD, and ABD. These columns are numbered 1, 2, 4, 8, 7, 11, 13, and 14.

Step 1: Calculate the W_i 's (i identifies the term in the RS). W_i is equal to the sum of all the weights associated with terms in the RS that contain the i^{th} term in the RS. Calculating the W_i 's determines which factors in the RS are most important. In the example problem, $W_1(A) = 119$, $W_2(B) = 117$, $W_4(D) = 118$, all others = 100.

Step 2: Rank order the first-order terms. The terms are ranked based on their W_i value, highest to lowest. This step ensures that the most important terms are assigned first and will not be confounded with any of the other highest ranked terms. A, D, B, are the three highest ranked, all others are equal and can be ranked in any order (i.e. A, D, B, C, E, F, G, H).

Step 3: Assign S variables by rank order. This step makes the first actual assignments. S terms can be assigned to the columns of the design as if the design were a full factorial design for S variables. An S-vector is used to identify the column assignments. The column assignment can be converted to an integer column number by treating the vector as a base 2 number. Appendix H presents the column representations for 16, 32, and 64 trial full factorial designs using numerical, S-vector, and alphabetical notations. For example, [0,0,1,1] is column 3. In this experiment, $S = 4$. Therefore the following assignments are made: A is assigned to column 1 (represented [0,0,0,1]). D is assigned to column 2 (represented [0,0,1,0]). B is assigned to column 4 (represented [0,1,0,0]). C is assigned to column 8 (represented [1,0,0,0]).

Step 4: Assign the interaction terms containing only the above variables. This step usually results in most of the interaction terms being assigned to columns. If all of the interaction terms are assigned in this step, then only first-order terms would remain to be assigned. First-order terms that do not appear in any interaction term can be assigned to any available column. The assignment of the interaction terms is done by using modulo 2 addition, element by element, on the two vectors that are associated with the interaction term. For example, AB is [0,0,0,1] + [0,1,0,0] mod 2. This yields [0,1,0,1] otherwise known as column 5. AD is [0,0,0,1] + [0,0,1,0] mod 2. This yields [0,0,1,1] otherwise known as column 3. BD is [0,1,0,0] +

$[0,0,1,0] \bmod 2$. This yields $[0,1,1,0]$ otherwise known as column 6. So far columns 1,2,3,4,5,6, and 8 have terms assigned to them.

Step 5: Assign the next variable by rank order. CAVE assigns the next variable to the column that results in the smallest increase in the objective function. The CAVE procedures evaluate assigning E to each column. Columns 7, 9, 10, 11, 12, 13, 14, and 15 are all equally good assignments (no new confounding is introduced). One of these choices is selected randomly. If there are g good assignments, then a $\text{INT}(g*U)+1$ will generate a uniform random number from 1 to g , where U is a uniform 0-1 random number and $\text{INT}()$ is an integer function that returns the integer portion of the operand. For this example, assume column 15 is selected.

Step 6: Any interaction terms associated with the latest assigned variable are now assigned as in step 4. Once a new first-order term is assigned, any interaction terms involving the newly assigned term and a previously assigned term must be assigned. In this example, no additional interaction terms remain to be assigned. Return to step 5 to assign the next ranking term.

Repeat step 5 (and step 6) for F, G, and H. For this example, assume F is assigned to column 14, G to column 11, and H to column 13.

Step 7: Print the solution. Step 7 is the end of the CAVE procedure. In this example, the assignments are A=1, B=4, C=8, D=2, E=15, F=14, G=11, H=13, AB=5, AD=3, BD=6.

4.3.2 ICAVE. The Iterative Column Assignment Via Examination (ICAVE) heuristic is a hybrid of a stochastic search procedure and the specialized structure and knowledge of the CAVE heuristic. It could be considered a local search heuristic in the region of the DOE/RS solution space centered about the design found using the CAVE heuristic. ICAVE uses the

same S-vector representation of column assignments to identify the specific column each term in the RS is assigned.

```
Procedure: ICAVE
begin
(1) Calculate  $W_i$ 's
(2) Rank variables by  $W_i$  in descending order
(3) Assign S variables by rank order
(4) Assign interaction terms
Do until all variables assigned
(5) Assign next variable by rank order
(6) Assign interaction terms
End Do
Do 100 times
(7) Unassign randomly selected first-order term (of rank
S+1 or greater) and associated 2fi terms
(8) Assign the first-order term as in step 5
End Do
(9) Print solution
end;
```

Figure 4-2. Pseudocode for ICAVE Heuristic

As seen in Figure 4-2, ICAVE modifies the CAVE heuristic by reassignment of the S+1 to K terms in the RS for a user-specified number of cycles. A term available for reassignment is assigned to each of the unassigned columns and the resulting design (and corresponding objective function value) is calculated. The term is reassigned to the column that results in the lowest objective function value. If two or more assignments result in the lowest objective function value, then the assignment is made randomly between these equally good candidate assignments just as in the CAVE procedures. In the implementation discussed here, the number of assessments was set equal to one hundred to balance solution quality and computational efficiency. Preliminary runs with the ICAVE heuristic indicated, that for some problems, eighty or more functional evaluations were performed before finding a zero cost solution. Therefore

choosing a number of functional evaluations smaller than eighty would decrease the expected solution quality in these experiments. If a zero-valued solution is found, at any iteration, however, the search terminates.

Future research could examine alternative termination criteria for the ICAVE heuristic. One candidate criteria is to terminate the search procedure when the change in the objective function value over some specified number of functional evaluations is less than ϵ (a small positive constant). This stopping criteria would permit the search to continue only as long as progress is being made towards the optimum solution. Such refinements could eliminate some of the computational inefficiencies in the heuristic tested in this research.

An ICAVE iteration evaluates all of the first-order terms in the RS once. One iteration is equivalent computationally to the CAVE heuristic. Figure 4-3 shows the distribution of ICAVE trials executed in solving the DOE/RS problem 32/25 presented in Chapter 3. Note that the problem has a zero-valued solution. A zero-valued solution allows ICAVE to terminate before the 100 iterations are executed if a zero-value solution is found first. Almost 40% of the trials resulted in ICAVE finding the optimal solution within the first ten iterations. Less than 10% of the trials required more than 90 iterations. Of course, ICAVE cannot terminate the search early for any problem without a zero-valued solution.

Whenever the procedure halted before completing the 100th iteration, ICAVE found an optimal solution. The search was also set to terminate once the 100th iteration was complete. The average number of functional evaluations for this problem was 32. The search found an optimal solution before terminating 96% of the time. This data suggested using 100 or so iterations as a stopping criterion for the ICAVE procedures. All of the 16-run experiments

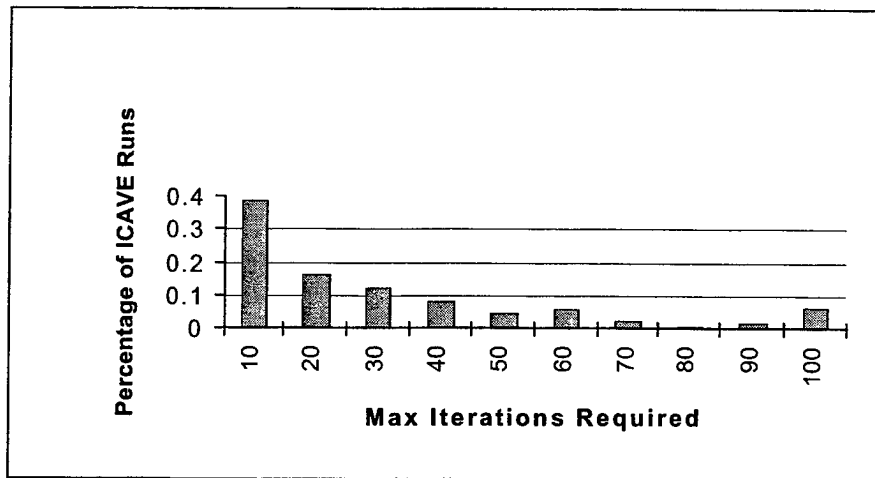


Figure 4-3. Functional Evaluations Required by ICARE in Solving DOE/RS Problem 32/25

executed the full 100 iterations. Clearly the stopping criteria for the ICARE procedure warrants further investigation.

When the CAVE heuristic assigns a term to a column, initially there are often many equally good choices. The variable is assigned to the first column that results in the best objective function value and the order in which the columns are assessed is random. By repeatedly examining the assignments (using the S-vector representation as described in step 5 for CAVE), first-order terms can be reassigned to new columns, which improves the overall design based on a complete assignment of all terms. Being assigned to a new, but equally good column, may allow some other term to be assigned to a new column, thereby generating an improved solution.

4.3.2.1 Solving an Example Problem Using ICARE. The ICARE procedures are the same as the CAVE procedures except that instead of terminating once all the factors in the RS have been assigned, ICARE attempts to improve the design by reassigning a randomly chosen factor to a different column. The ICARE procedures begin by repeating the CAVE procedures. For

the ICAVE example problem, ICAVE performs the same initial procedures as in the CAVE example until all the terms in the RS have been assigned to a column. Step 7 for ICAVE is to unassign a randomly selected first order term (ranked from $S+1$ to K) and the associated interaction terms.

Section 4.3.1.1 solved an example problem using CAVE. The solution to the problem was $A=1$, $B=4$, $C=8$, $D=2$, $E=15$, $F=14$, $G=11$, $H=13$. Continuing this example with the subsequent ICAVE procedures, assume that factor H was randomly chosen for reassignment to a different column. ICAVE keeps the assignment of all the terms not associated with H constant. Step 8 then examines the value of assigning H to each column. In our example, assigning H to column 7 results in a known minimum aberration design. $Y=0$ and the objective function value is zero. Since zero is the best possible value for the DOE/RS objective function, the search terminates with a known optimal solution.

4.3.3 Characteristics of the CAVE and ICAVE Heuristics. The CAVE heuristic has the advantage of quickly finding a feasible initial solution. The ICAVE heuristic increases the likelihood of finding a better solution but at the expense of increased computational effort. If the CAVE method finds a solution that meets the user's needs, then the other, more time consuming, methods need not be used. Additionally, the solution found by CAVE (or ICAVE) can be used as the starting point for the other methods in this study (the simulated annealing approach to be developed later in this chapter used CAVE for an initial starting solution) or developed for follow-on work.

4.3.4 The CAVE/ICAVE Experiment. The CAVE/ICAVE experimental design was the same as the IP design, except that repetitions were conducted. The DOE/RS problems were

solved an average of more than five times using each heuristic method; additional runs were used to achieve a better estimate on the experimental designs with the greatest variances in the ICAVE performance. The results of these experiments are presented in the following subsection.

4.3.5 The CAVE/ICAVE Results. Table 4-1 presents the results of the CAVE experiments. As shown in Table 4-1, CAVE found solutions whose relative reduction metric ranged from 89% to 100%. The average CAVE performance over the three sizes of DOE/RS problems was 96% (16 design points), 92% (32 design points), and 91% (64 design points), with an overall average of 93%.

TABLE 4-1: THE OVERALL CAVE PERFORMANCE

ID	Mean Objective Function (OF)	Min OF	Max OF	% Relative Reduction	% Relative Reduction Std Dev	Evaluations
16/11	0	0	0	100.0	0.00	38
16/12	17	17	17	100.0	0.00	38
16/13	24	18	27	99.1	0.00	38
16/15	116	116	116	90.4	0.00	38
32/25	93	0	195	93.9	0.00	161
32/28	139	36	198	91.3	0.00	161
32/31	169	87	281	90.1	1.37	161
64/51	320	170	407	89.4	0.00	525
64/57	440	183	718	90.5	1.67	525
64/63	784	558	1139	92.6	1.24	525

The CAVE heuristic consistently found the optimal solution for the 16/12 problem and often found the optimal solution for the 16/13 problem. A 96% performance for experiments of 16 runs suggests that CAVE may be particularly effective for smaller experiments.

Table 4-2 presents the results of the ICAVE experiments. The ICAVE heuristic performed consistently well over all nine problems averaging a solution quality of 99.4%. The lowest average percentage of best solution found was 97.58%, while the largest standard deviation for the percentage of best solution found was 1.67%. The ICAVE heuristic demonstrated highly consistent performance. ICAVE had an average performance over the three sizes of experiments of 99.96%, 99.34%, and 98.80%. The time required to execute the CAVE and ICAVE searches should easily support most experimental objectives, especially considering the rapid improvements in computing technology enjoyed today.

TABLE 4-2: THE OVERALL ICAVE PERFORMANCE

ID	OF	Min OF	Max OF	% Relative Reduction	% Relative Reduction Std Dev	Time (Pent 100)	Evaluatio ns
16/11	0	0	0	100.0	0.00	1 min	38
16/12	17	17	17	100.0	0.00	5 min	14,400
16/13	18	18	18	99.9	0.00	7 min	14,400
16/15	41	41	41	100.0	0.00	8 min	14,400
32/25	0	0	0	100.0	0.00	3 min	67,200*
32/28	0	0	0	100.0	0.00	12 min	67,200*
32/31	34	0	66	98.0	1.37	18 min	67,200*
64/51	0	0	0	100.0	0.00	135 min	211,200*
64/57	173	135	221	98.8	1.67	181 min	211,200
64/63	578	550	606	97.6	1.24	210 min	211,200

As seen in Table 4-3, the ICAVE performance is a material improvement over the CAVE average of 93.0%. ICAVE utilizes approximately 400 times the computational effort of CAVE for an average improvement of 6.4% to an outstanding 99.4% of optimal. The improvement is only 3.6% on the 16 run experiments because of CAVE's strong performance on the smaller experimental design and the nature of the percentage of the best-known solution as a metric.

TABLE 4-3: CAVE VS. ICAVE PERFORMANCE

ID	CAVE % Relative Reduction	ICAVE % Relative Reduction	Δ %Relative Reduction	ICAVE Evaluations/ CAVE Evaluations
16/11	100.0	100.0	0.00	1
16/12	100.0	100.0	0.00	379
16/13	99.1	99.9	0.75	379
16/15	90.4	100.0	9.63	379
32/25	93.9	100.0	6.10	417
32/28	91.3	100.0	8.66	417
32/31	90.1	98.0	7.98	417
64/51	89.4	100.0	10.58	402
64/57	90.5	98.8	8.29	402
64/63	92.6	97.6	4.97	402

Detailed experimental results are provided in Appendix B. The software code and user instructions for the CAVE and ICAVE heuristics are presented in Appendix E.

CAVE and ICAVE appear to be a highly effective heuristics. Ultimately, their value depends largely on how well they compare to standard general-purpose heuristics in solving the DOE/RS problem. Section 4.4 presents the answer in the form of a simulated annealing heuristic applied to the DOE/RS problem.

4.4 Simulated Annealing

Simulated annealing serves as a basis for evaluating the CAVE and ICAVE heuristics. For ICAVE to be a truly useful heuristic, it needs to perform as well or better than a general purpose heuristic utilizing the same computational resources. This section presents a simulated annealing heuristic developed to solve the DOE/RS problem: Design of Experiments using Simulated Annealing (DESA).

Simulated annealing (SA) was first introduced by Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller, (1953) to simulate atoms undergoing the annealing process. In the physical act of annealing, a solid is heated to almost melting and slowly cooled. If the solid is cooled slowly enough, the atoms arrange themselves in a lowest potential energy state. When simulating the annealing process, the objective function represents the potential energy of the atoms. The simulation attempts to find the configuration (or solution) which corresponds to the minimum energy state.

SA essentially is a biased random walk that attempts to locate a global optimum without becoming trapped in a local optimum. Simulated annealing mimics the physical annealing process to direct a mathematical search for optimal solutions. In SA, a solution to the problem to be optimized corresponds to the configuration of atoms in the solid to be annealed. The objective function corresponds to the potential energy of the atoms. The solution randomly changes as a control parameter analogous to temperature (and called temperature in this study) is slowly decreased. Just as the simulated atoms arrange themselves in the minimum energy state, the final solution of the SA procedures is ideally a solution corresponding to a global minimum solution.

SA has been used to solve various optimization problems by Kirkpatrick, Gelatt, and Vecchi (1983), Bononi and Lutton (1984), Cerny (1985), Bohachevsky, Johnson, and Stein (1986), Haines (1987), Conoly (1992), Deutsch (1992), Cardoso, Salcedo, and de Azevedo (1994), Heine (1994), Winston (1994), Woodruff (1994) as well as others. Collins, Eglese,

and Golden (1988) provide an excellent annotated bibliography of the initial development of SA.

SA begins with a current solution and then perturbs that solution to generate a candidate solution. If the candidate solution is more desirable than the current solution, the candidate solution becomes the new current solution. Under certain conditions (discussed later) the SA will accept the candidate solution even if it is not as desirable a solution as the current solution. This behavior allows the SA procedures to escape local optima in an attempt to find a global minimum. SA's ability to find global optimum and near-optimum solutions in large problems with a number of local optima makes SA an approach of interest to operations research practitioners (Bohachevsky, Johnson, and Stein, 1986, p. 209).

4.4.1 SA Applications. Both Bohachevsky, Johnson, and Stein (1986) and Haines (1987) have applied SA to solve Design of Experiments problems. Each used SA to select the levels of experimental designs to construct D-optimal designs; that is, they minimize the determinate of the $X'X$ matrix, where X is the experimental design matrix and X' is the transpose of the design matrix. Haines extended this research to include I-optimal and G-optimal designs as well. Their works suggest that SA may be successfully applied to other DOE problems.

Because of the promise shown by these researchers in using SA to solve DOE problems, an SA heuristic was chosen as a potential alternative to the newly developed heuristics of CAVE and ICAVE presented earlier.

This area of research was initially motivated by a specific design problem encountered while using Response Surface Methodology (RSM) techniques to study large combat models

(Forsythe 1994). An initial SA code was written that provided a design that enabled the estimation of all the terms in the RS. The problem was a 2^{9-4} fractional factorial design with three interaction terms which were known to be significant from a prior experiment, and 15 interaction terms which were considered likely to be important in the Response Surface model. The remaining interaction terms had either been shown to be immaterial or were theoretically unrelated to the study objectives and assumed to be insignificant.

4.4.2 Characteristics of SA. Lam suggests that SA is an excellent choice for a solution methodology when high quality solutions are desired, large amounts of computational time can be afforded (at least thousands or even tens of thousands of solution evaluations), and the problem has many degrees of freedom (Lam 1988, p. 164). DOE/RS problems exhibit all of these characteristics. A SA approach to solving the DOE/RS problem appeared promising and is especially useful for designing experiments with large numbers of variables or interactions. The fact that the objective function when solving a DOE/RS problem is a nonnegative function has the potential to help avoid the excessive computational efforts often associated with SA, by allowing the procedure to terminate if a zero-valued solution is found.

Although SA gives the experimenter the ability to rapidly generate promising designs, a weakness of the SA approach is that if it terminates with a nonzero-valued solution, it cannot be established with certainty that a better solution does not exist. This uncertainty of the optimality of the solution is a weakness of all heuristic methods. Both researchers and practitioners prefer an approach (such as an IP model) that finds optimal solutions in a timely fashion to heuristics, if such an approach is available.

```

Procedure: SIMULATED_ANNEALING
Begin
(1) INITIALIZE
(2) Perturb current solution to generate a trial solution
(3) Calculate the objective function value for the trial solution ( $z_{temp}$ )
(4) If  $z_{temp} < z_{cur}$  then trial solution becomes new solution
    else if  $\exp^{((z_{cur} - z_{temp})/t)} > \text{uniform}(0,1)$ , then trial solution becomes
        new solution
(5) Adjust  $TLcounter$ 
(6) If  $TLcounter = TL$  then  $t = (CM)t$ 
(7) If stop criteria met then end
Go to Step (2)
end;

```

Figure 4-4. Pseudocode for a Simulated Annealing Algorithm

4.4.3 SA Structure. The SA procedures used in this research correspond to the “standard” homogeneous implementation of SA presented by Laarhoven and Aarts (1987, Ch. 2). While variations of SA exist, these standard procedures performed well in preliminary studies. The specific SA code used can be found in Appendix F. Figure 4-4 is a pseudocode description of the SA procedures adapted from Laarhoven and Aarts (1987, p. 10). Key implementation decisions are discussed later in this chapter.

The SA procedures have five components: the problem depiction, the permutation mechanism, cooling schedule, acceptance function, and stopping criteria. The *problem depiction* consists of a cost function (objective function) and an initial configuration or starting solution (from the solution space). For each problem there must be a *permutation mechanism* that provides a means of generating a local permutation of the current solution to generate an alternative solution. The SA procedure calculates the cost associated with the perturbed solution relative to the unperturbed solution and selects, via an acceptance function, one solution to continue as the new current solution.

A *cooling schedule* is required to control the rate of decrease in the control parameter, “temperature.” The temperature parameter is analogous to temperature in an annealing process. The higher the temperature, the more motion of the entities being optimized. An *acceptance function* chooses between a current solution and a candidate solution. Finally, *stopping criteria* are necessary to halt the search.

4.4.3.1 Problem Depiction. The SA heuristic represents the DOE/RS problem as an assignment of the first-order terms of the RS to columns in the design matrix.

The objective function is the same as equation 3-1 presented in Chapter Three and for the CAVE and ICAVE heuristics:

$$\min z = \sum_{i=1}^N C_i x_i + C_{\min ab} y \quad (4-1)$$

4.4.3.2 Perturbation Mechanism. SA perturbation mechanisms are problem specific. The perturbation mechanism must alter an existing solution to form a new solution to the problem under study. The perturbation mechanism used in this research creates a new design by changing the column assignment of one of the first-order terms in the RS. A first-order term of the RS is selected at random. The current assignment is perturbed by selecting one element from the vector that represents the current solution and changing its value from zero to one or from one to zero. In the case of the ABD column (1,0,1,1) in a 2^4 design, the new column would be either BD (1,0,1,0), AD (1,0,0,1), ABCD (1,1,1,1), or AB (0,0,1,1). If another first-order term was previously assigned to the new column, then this first-order term is reassigned to the

original column. Simply put, the two first-order terms exchange column assignments, thus avoiding confounding any first-order terms.

4.4.3.3 Cooling Schedule. In SA, *temperature* (t) is a term for a control variable that performs a function analogous to thermal temperature in the annealing process. The temperature determines how biased the SA procedure is towards accepting only better solutions. The probability that a worse solution will be accepted as the new current solution is determined by an acceptance function. The Metropolis criterion simulates the statistical behavior of annealing and is the most widely used acceptance function (Sasaki and Hajek, p. 388). The Metropolis criterion is $e^{-\Delta z/t}$, where Δz is the difference in the objective function values ($z_{\text{cur}} - z_{\text{temp}}$), and t is the current temperature. Other acceptance functions have been studied. These other functions have not exhibited significant improvement to the Metropolis approach. Readers interested in other acceptance functions are directed to the work of Anily and Federgruen (1987, pp. 657-667).

The larger the Δz and the smaller the t value, the less likely the search procedure will accept the perturbed solution as the new current solution. In most SA procedures, a zero-to-one uniform random variable, $U(0,1)$, is generated (Ingber 1993). If the $U(0,1)$ random draw is less than $e^{-\Delta z/t}$, then the candidate solution becomes the new current solution.

A cooling schedule is used to control the rate at which the temperature is reduced. Tailoring the parameters of the cooling schedule is usually necessary for the SA algorithm to run efficiently. Selection of the various SA parameters is problem specific and typically done through empirical data (Kirkpatrick, Gelatt and Vecchi 1983, p. 679 and Ingber 1993). The

cooling schedule utilizes three parameters: the initial starting temperature, the decrement function, and the final stopping temperature.

4.4.3.3.1 Starting Temperature. The temperature parameter controls the likelihood of accepting a worse solution as the next current solution. Higher “temperatures” are more likely to cause the search procedures to move in a nonimproving direction just as a higher temperature will make a group of atoms more likely to occupy a higher energy configuration. The ability to move in nonimproving directions enables the SA procedure to escape local minima while searching for a global optimum.

Research has shown that the starting temperature has a significant effect over the performance of the SA heuristic (Boese, Kahng, and Tsao 1992, p. 22). If the starting temperature is too high, a great deal of computational time will be utilized in what is for the most part a totally random walk through the solution space. If the starting temperature is too low, then the search space may be restricted because the search procedures will be less likely to escape a local minimum. Many procedures attempt to start at a temperature where the acceptance rate (i.e., the ratio of total perturbations/perturbations accepted as a new solution) is at least 50% (Aarts and Korst 1989, p. 16, Kirkpatrick, Gelatt and Vecchi, 1983, p. 675).

The desired starting temperature can be calculated if the maximum and minimum possible objective function values are known. The initial temperature may then be selected such that the acceptance function has at least a 50% chance of accepting any perturbed solution when compared with any current solution. Since the probability of a worse solution being accepted is equal to $e^{(-\Delta Z/T)}$, if $e^{((z_{min} - z_{max})/t)}$ is greater than or equal to 0.5, then any new solution which is

worse than the current solution must have at least a 50% chance of being accepted as the new solution.

The largest possible Δz for a given DOE/RS problem is equal to the sum of all the terms in the objective function. The initial temperature can then be set so that the probability of accepting a perturbed solution is at least 50% (i.e. $\exp(-Z_{\max}/T_{\text{start}}) = 0.5$). Using this starting temperature guarantees that the initial acceptance criteria has at least a 50% chance to accept any new solution. In practice, the expected acceptance rate would be much higher since most solutions have neither the maximum nor minimum values. This approach establishes an initial temperature sufficient to allow the search to easily escape local minima. Using the largest CAVE solution value instead of the largest possible Δz would provide an acceptable alternative while minimizing the risks of a pure random walk.

4.4.3.3.2 Temperature Length. Temperature length (TL) is the number of iterations that the search procedure stays at one temperature. Theoretically, if the search procedures stay sufficiently long at each temperature, then an optimal solution is guaranteed (Hajek 1988, p. 315). Unfortunately, one would need an infinite amount of computational time to be certain that sufficient iterations had been performed at each temperature. SA procedure implementations often identify the ideal TL values empirically. Others define a number of acceptances that will indicate sufficient equilibrium (Laarhoven and Aarts 1987, Ch. 5). Variations on TL include (Collins, Eglese, and Golden 1988):

- Constant: $TL = CM$, where CM (*Cooling Multiplier*) is defined in 4.3.3.3
- Arithmetic: $TL_i = TL_{i-1} + C$, where C is an integer constant
- Geometric: $TL_i = CM * TL_{i-1}$

- Acceptances of perturbed solutions: Repeat until a number of new solutions have been accepted.
- Rejections of perturbed solutions: Repeat until a number of new solutions have been rejected.

Again, waiting too long before reducing the temperature can waste computational effort.

Fortunately, parametric experimentation will generally indicate good settings for a TL as well as the sensitivity of the SA procedure to variations in TL . Establishing a robust TL for the DOE/RS problem is a subobjective of this study. A “robust” TL is one that allows the SA search to find near optimal answers quickly across a variety of DOE/RS problems.

Preliminary research indicated sampling more than 150 times per temperature change did not yield significant improvement in the resulting solution quality. Therefore, $20*TL$ perturbations (50-150 perturbations in this research) were used. This research utilized a combination of temperature length mechanisms one and four. Once $20*TL$ perturbations occurred or if TL of the solutions were accepted, then the temperature function was decreased. In either case, the t was decreased by multiplying the current temperature (t) by the cooling multiplier (CM). This research evaluated CM values of 0.9, 0.93, and 0.95.

4.4.3.3.3 Cooling Rate. The cooling rate is the change in temperature over time. Cooling too slowly causes the procedures to spend unnecessary time searching.

The selection of the “cooling” parameters has an impact on the quality of solutions that the SA algorithm obtains. In general, a slower schedule (i.e. one that starts from a higher temperature t , has a larger cooling multiplier CM , and has a larger temperature length TL) will lead to better solutions. On the other hand, slower schedules tend to use more CPU time.

Recall that high temperatures allow acceptance of worse solutions, which, if done unnecessarily, wastes computational effort. Cooling too quickly causes the SA procedures to move only in improving directions. Without the ability to consider inferior solutions as the new current solution, SA would have no way to escape a local minimum. Therefore cooling too quickly increases the likelihood of the search becoming trapped in a local minimum. Johnson states that geometric cooling schedules of the form $t_{i+1} = (CM)t_i$ are effective and appear to be the preferred cooling schedule found in the literature (Johnson, Aragon, McGeoch, and Schevon, 1991, p. 380). The cooling multiplier, CM , is often a constant just less than one. Collins, Eglese and Golden (1988, p. 2) state that, in their survey of SA, typical values for CM are 0.99, 0.95 or 0.90.

4.4.3.4 Acceptance Function. z_{cur} is the objective function value associated with the current solution. z_{temp} is the objective function value associated with the perturbed solution. The acceptance function utilized in this research is the standard exponential form: $\exp((z_{cur}-z_{temp})/t)$ when z_{temp} is greater than z_{cur} . The perturbed solution is always accepted as the new solution if z_{temp} is less than or equal to z_{cur} .

Figure 4-5 is a graph of $\exp((z_{cur}-z_{temp})/t)$ versus $(z_{cur}-z_{temp})/t$. This graph shows the effect of temperature on the probability of accepting various solutions which are worse than the current solution.

When the temperature is equal to Δz , there is a 37% chance of moving in a nonimproving direction. As the temperature approaches zero, the SA procedures will be unlikely to escape from a local minimum.

4.4.3.5 Stopping Criteria. Every search methodology needs criteria for ending the search.

The termination conditions represent a tradeoff between the burden of additional computational effort and the probability of finding a better solution. If a known optimal solution has been

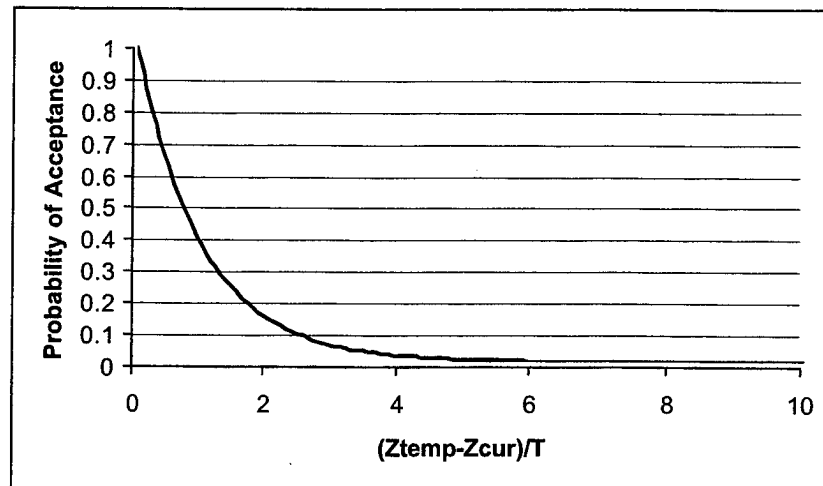


Figure 4-5. The Effect of Temperature on SA Behavior

found, then clearly the search should be terminated. Unfortunately, it is often not possible to tell if a given solution is optimal unless the candidate solution can be shown to satisfy some specific optimality conditions. Much of the difficulty found with applying SA to combinatorial optimization problems is the tendency for the search to spend major portions of its computational effort at low temperatures in searches that often do not improve the final solution.

It has been shown in Chapter Three that the DOE/RS problem does have a criteria that is sufficient to guarantee optimality. The best possible solution of the DOE/RS problem, if it exists, is a fractional factorial design which allows a clear estimate of the effect of each term of the RS (i.e. $z = 0$). If such a solution can be found, then the SA search procedures can be

terminated. If such a guaranteed optimal solution is not found, or is not feasible for a particular requirement set, then another termination criteria is used.

Common termination criteria in SA include (Collins, Eglese, Golden 1988):

- *Iterations*: Terminate after a fixed number of iterations. This is most useful when the available computational resources are known and limited.
- *Temperature* (T_{stop}): Terminate when the temperature reaches some predefined temperature (usually near zero).
- *Objective Function Value*: Terminate when the objective function value does not materially change (e.g. greater than ϵ , a constant) after some defined search duration. This termination condition requires recording the objective function value over a specified search duration. This search duration could be defined in terms of change in temperature, number of iterations, or even real time. If the difference between the maximum and minimum z values is less than ϵ , then the search is terminated. This method could result in ending the search prematurely.

For some experimental designs, an experimenter might wish to specify a value near zero that would constitute a satisfactory design. For example, an experimenter might specify a RS including four nuisance terms. Nuisance terms are terms that the experimenter is not specifically interested in estimating, but does not wish to confound with any other terms in the RS. Let each nuisance term be assigned a value of one (and all the more significant terms have values greater than four). Confounding among the nuisance terms would be acceptable as long as no other confounding exists. For such an experimental design, a solution value of four or less would be considered acceptable. The acceptability of neglecting those nuisance terms would be left to the judgment, experience, and knowledge of the experimenter.

- *Acceptance*: The search is ended when the ratio of acceptances to perturbations falls below a defined criterion.

In solving the DOE/RS problem, the search can terminate if a zero-valued solution is found; such a solution satisfies the RS. Otherwise, the DESA heuristic continues searching until the stopping temperature (T_{stop}) is reached. Setting T_{stop} too low could result in spending time unproductively when the search procedure is unlikely to find a better solution. Setting T_{stop} too high could result in failure to find an improved solution. This research investigated T_{stop} values of 20, 10, and 0.1.

The software code implementing the DESA heuristic is presented in Appendix F.

4.4.3.6 An Example Problem. The following example uses the Bisgaard and Fuller problem presented in Appendix C to illustrate the SA procedures. The RS consisted of (A, B, C, D, E, F, G, H, AB, AD, and BD). The experiment was a 16-run experiment. The weights assigned to each of the variables are the same as the weights in the example IP problem: 100 for each of the first-order terms, 10, 9, and 8 for the interaction terms, and 500 for C_{minab} . The methodology will consider only the specified minimum aberration designs if a C_{minab} value much larger than the sum of all other coefficients is used.

The initial assignment used for this problem was: $A = [1, 0, 0, 0]$, $B = [0, 0, 1, 0]$, $D = [0, 1, 0, 0]$, $C = [0, 0, 0, 1]$, $E = [1, 1, 1, 1]$, $F = [0, 1, 1, 1]$, $G = [1, 1, 0, 1]$, $H = [1, 0, 1, 1]$. This solution has a value of 500.

The SA procedures begin by selecting a term, in this case let it be E. The SA approach then randomly selects one of the four elements of the assignment vector and changes it. Assume that the fourth bit was changed to a zero in this example. This creates a new design where E is

assigned to [1,1,1,0] or column seven. The objective function value of this design is zero. This is better than the current design and so it becomes the new design. It is also better than the best design so far and so this data is stored as the best design. In this way, the best design ever seen while searching is presented to the experimenter. A zero-valued solution is also known to be optimal, so this search would terminate with an optimal solution.

If the new solution had not been a zero-valued solution, the search would have continued by perturbing the new current solution and comparing the two designs. If the new solution were worse than the current solution, the probability that the new solution is accepted as the new solution is given by the Metropolis criterion: $\exp((z_{\text{cur}} - z_{\text{temp}})/t)$. The temperature is reduced on a parametrically defined schedule. The SA process continues until a zero-valued solution was found or until the stopping temperature is reached.

4.4.4 The SA Parameter Experiment. In any SA application, it is necessary to examine the impact of key parameters on the performance of the heuristic. A 3^{5-2} experimental design was selected to examine the effects of S , $\%UC$, TL , CM , and ST on the SA performance. The design was constructed by repeating a 3_{III}^{4-2} design (Montgomery, 1999 p. 460) for each of the three values of S . The levels chosen for the TL , CM , and ST were based on a pilot study (Forsythe, Auclair, Deckro, and Shedden, 1997) and are listed in Table 4-5. The pilot study determined that values beyond the specified ranges for TL , CM , and ST did not improve the performance of the pilot SA heuristic. These settings are also within the range of settings used by SA heuristics found in the literature (Ingber 1993). The design allows the estimation of the main effects of each of the experimental factors while utilizing one-ninth of the runs required for a full factorial design.

The Percentage of Unassigned Columns ($\%UC$) indicates how close the required DOE/RS problem is to being completely saturated. The DOE/RS problems solved in this experiment are the same as presented in Chapter 3. For example, $S = 4$ and $\%UC = 0$ is DOE/RS problem 16/15. $S = 6$ and $\%UC = 20$ is DOE/RS problem 64/51.

Ten repetitions were performed for each of the nine DOE/RS problems. The number of repetitions was selected based on the observed distribution of the objective function found in the preliminary SA data. The confidence interval width of any term estimated via regression is a function of the number of observations. Increasing the number of experimental runs reduces the confidence interval.

Unfortunately the width of the confidence interval is proportional to the inverse square root of the number of samples. Ten iterations of the SA experimental runs appeared to represent an acceptable tradeoff between improving the estimates of the effect and the computational effort expended.

The overall SA design is as presented in Table 4-4. As was discussed earlier, S and $\%UC$ are characteristics of the DOE/RS problem space. There were two conditions used as stopping criteria. If a zero-valued solution was found, it was known to be optimal and the search terminated. If a temperature less than the stopping temperature (ST) was reached, then the search was terminated.

The results of this experiment provide information on the impact of the SA parameters over the range of DOE/RS problem space.

TABLE 4-4: OVERALL SA DESIGN

Run	S	TL	CM	ST	%UC
1	4	5	.90	0.1	0
2	4	5	.93	10	10
3	4	5	.96	20	20
4	4	10	.90	10	20
5	4	10	.93	20	0
6	4	10	.96	.01	10
7	4	15	.90	20	10
8	4	15	.93	.01	20
9	4	15	.96	10	0
10	5	5	.90	0.1	0
11	5	5	.93	10	10
12	5	5	.96	20	20
13	5	10	.90	10	20
14	5	10	.93	20	0
15	5	10	.96	.01	10
16	5	15	.90	20	10
17	5	15	.93	.01	20
18	5	15	.96	10	0
19	6	5	.90	0.1	0
20	6	5	.93	10	10
21	6	5	.96	20	20
22	6	10	.90	10	20
23	6	10	.93	20	0
24	6	10	.96	.01	10
25	6	15	.90	20	10
26	6	15	.93	.01	20
27	6	15	.96	10	0

4.4.5 The Parameter Experimental Result. This section summarizes the experimental results of the DESA parameter experiment. Detailed experimental results are also provided in Appendix A for the DESA heuristic

The effects of the SA parameters on the quality of the solution and the computational effort expended can be seen in Figures 4-6 and 4-7.

In Figures 4-6 and 4-7, the diamond indicates the median result for each parameter setting. The minimum and maximum values are indicated by “plus” sign. In three of the DOE/RS problems an optimal solution was not determined because of computational limits; the

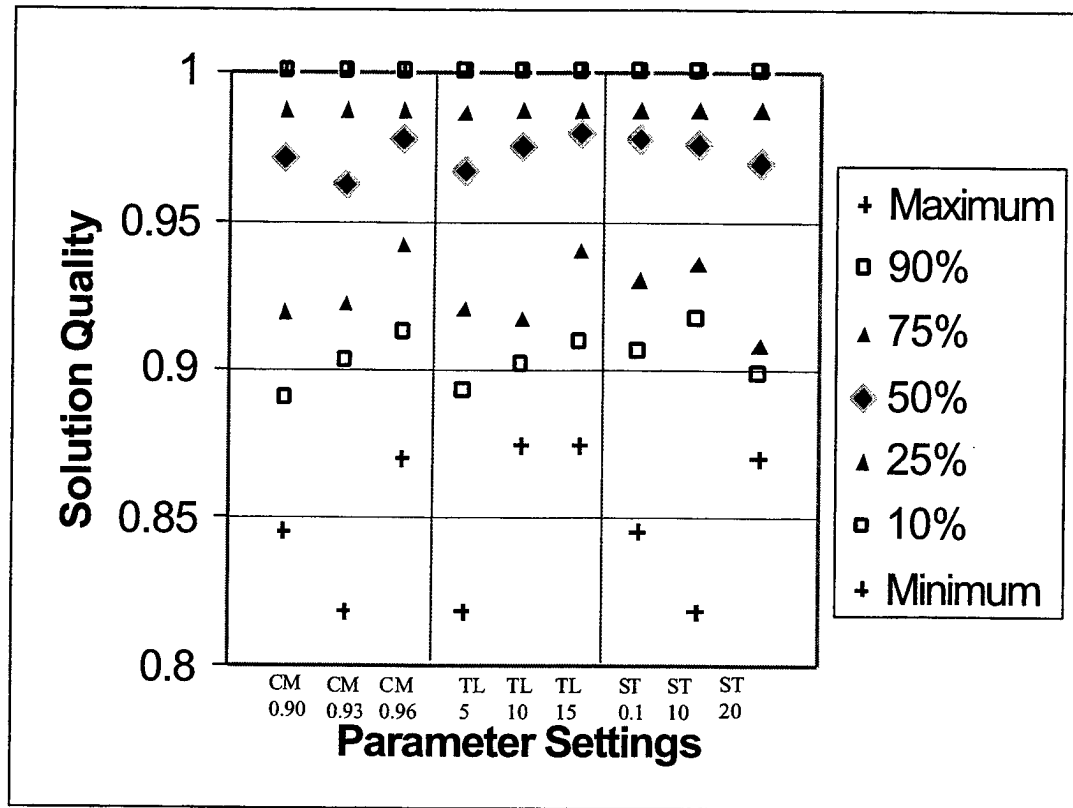


Figure 4-6. The Effect of SA Parameters on Solution Quality

best known solution is used as a substitute for comparison purposes. The standard deviations were calculated by estimating the variance of the solution value at each data point.

Figure 4-6 highlights the consistency of performance of DESA in terms of solution quality. This consistency suggests that DESA's performance is relatively insensitive to changes in its parameters, thus supporting the idea that consistent DESA performance can be expected from a standard set of parameters for DOE/RS problems.

As seen in Figure 4-7, all three SA parameters (the cooling multiplier, the temperature length, and the stopping temperature) affected the number of evaluations performed by DESA.

The following statistic was used to evaluate the relative effectiveness of each of the parameter settings: $(\text{maximum evaluations} - \text{minimum evaluations}) / \text{minimum evaluations}$. *Maximum evaluations* are the largest median number of evaluations associated with a parameter. The *minimum evaluations* are the smallest median number of evaluations associated with each parameter. This statistic measures the relative impact of selecting the best (of the three) settings versus selecting the worst.

Of the three parameters, the stopping temperature had the most dramatic effect: $(12060 - 2056) / 2056$ or 487%. Since the stopping temperature was the primary termination criterion, it was expected that the number of runs would increase as the stopping temperature decreased. The cooling multiplier also played an important role in determining the number of iterations DESA performed: $(10600 - 3714) / 3714$ or 185%. The cooling multiplier data suggests a nonlinear effect with a minimum near 0.93.

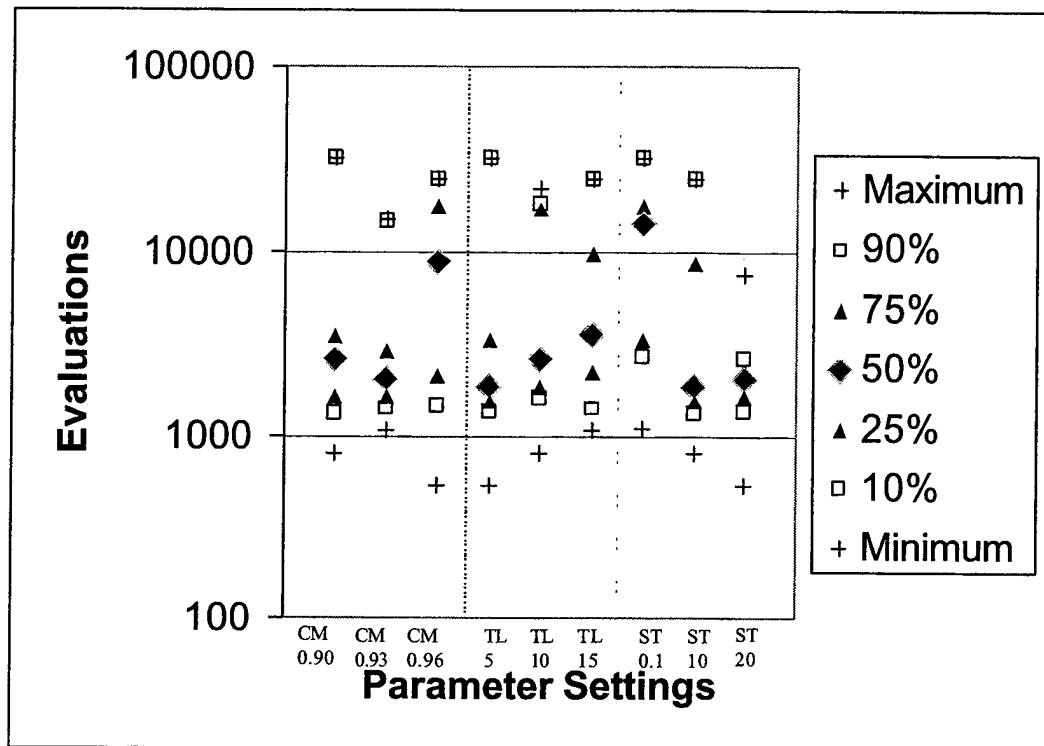


Figure 4-7. The Effect of SA Parameters on Computational Effort

In contrast, the temperature length statistic is only $(7683-5380)/5380$ or 43%. If 15 acceptances always occurred before 100 perturbations, changing TL from 5 to 15 would have had no effect. If TL acceptances never occurred before $20*TL$ perturbations, then the TL effect on evaluations would have been 200%. The data in Figure 4-7 suggests that temperature was reduced due to the number of TL acceptances more often than the number of perturbations.

Of the three cooling multipliers, 0.93 had the lowest evaluation count. Cooling at a rate of 0.9 caused the search to generate a higher rejection to acceptance ratio that caused an

increased number of evaluations. Again, supporting the literature's finding, the larger the cooling multiplier, the slower the temperature approaches the stopping temperature.

Considering these results, the median levels for the DESA implementation of a DOE/RS problem provide reasonable performance and computational effort without risking operating at the extremes of the experimental region, and avoid potential interaction effects. The estimated performance for these parameters was 6,000 evaluations and a relative reduction measure of 95%. DESA runs were conducted using these levels to compare with the estimated performance. The results confirmed the validity of the results of the parameter experiment. 6400 evaluations were used on average resulting in a solution quality of 97%. These levels were used to evaluate DESA's performance in solving the ten DOE/RS problems.

From my perspective, the settings that optimize solution quality tend to maximize the number of evaluations. The parameters in the center of the design seem to offer a reasonable balance between performance and effort. Such an assessment is partially subjective and therefore other researchers might have chosen differently based on the available data and their experiences and preferences. The results of this evaluation are presented in the following section.

4.4.6 The DESA Evaluation Experimental Result. Once parameters for the DESA heuristic had been established, it was possible to measure the effectiveness and efficiency of DESA to solve the nine DOE/RS problems.

The mean DESA heuristic's performance data is presented in Table 4-5. The average result was 96% of the optimal solution value, while the average of each design point exceeded 90% of the optimal value. Table 4-6 represents the average of all the test points.

TABLE 4-5: THE OVERALL DESA PERFORMANCE

ID	Mean Objective Function (OF)	Min OF	Max OF	% of Relative Reduction	OF Std Dev	Mean Evaluations	Evaluations Std Dev
16/11	0	0	0	100	0.0	1183	625
16/12	27	27	27	98.7	0.0	2180	982
16/13	27	27	30	98.7	0.1	7975	9606
16/15	51	51	54	99.9	0.1	19370	12950
32/25	13	0	48	99.1	1.2	2098	2302
32/28	61	0	132	96.2	2.3	6445	7032
32/31	138	35	261	93.8	3.3	4403	2671
64/51	226	111	393	92.5	2.3	7149	5542
64/57	442	264	718	90.5	2.9	5353	6270
64/63	693	500	851	94.0	2.9	5342	3043

The median performance of the DESA heuristic is shown in Table 4-6. The change in Relative Reduction column shows that CAVE outperformed DESA on problems 16/12 and 16/13. DESA outperformed CAVE on problems 16/15 through 64/63.

TABLE 4-6: THE MEDIAN DESA PERFORMANCE

ID	OF	CAVE % Relative Reduction	DESA % Relative Reduction	ICAVE %Relative Reduction	Median Evaluations
16/11	0	100.0	100.0	100.0	722
16/12	27	100.0	98.7	100.0	1626
16/13	27	99.1	98.7	99.9	1417
16/15	51	90.4	99.9	100.0	24570
32/25	0	93.9	100.0	100.0	1747
32/28	51	91.3	96.9	100.0	2081
32/31	133	90.1	94.1	98.0	3308
64/51	216	89.4	92.9	100.0	4542
64/57	429	90.5	90.9	98.8	2995
64/63	698	92.6	93.9	97.6	3494

Combining information about both the median and the mean number of evaluations gives a better insight into the DESA results. For example, in the second problem (16/13), the mean

number of evaluations is 7975 compared to 1417 as the median, which suggests a strongly asymmetrical distribution.

When the median value is much smaller than the mean value of a statistic, the distribution is skewed, indicating one or more of the runs were significantly longer than the others. Figure 4-8 may provide an even better understanding of the DESA performance distribution. Figure 4-8 is a histogram of DESA performance over all nine DOE/RS problems.

Performance is measured as the ratio of the value function of each run's solution compared to the value of the Relative Reduction metric for that DOE/RS problem. Equation 3-31 defines the performance metric. DESA solutions were within 20% of the optimal solution

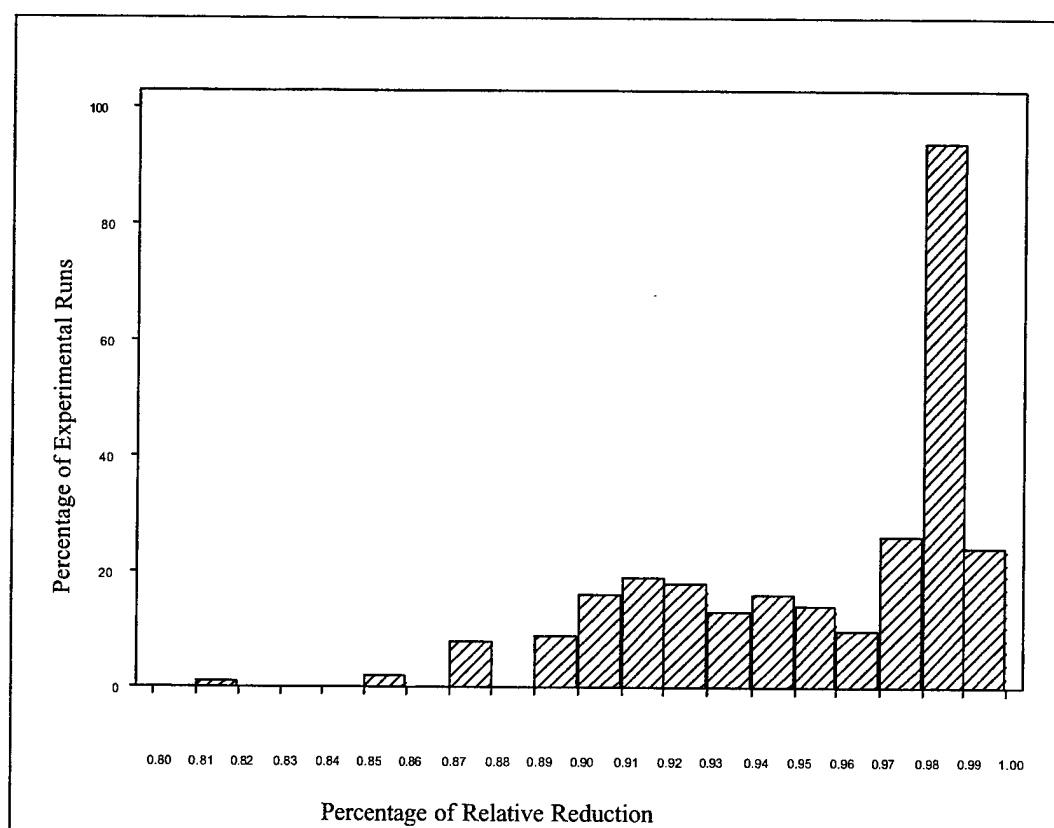


Figure 4-8. Histogram of DESA Effectiveness

value 100% of the time. Notice that the DESA procedures achieved at least 90% of the optimal solution 92.6% of the time.

An average DESA run took on the order of 10 minutes to run on a Pentium 100. Computers continue to increase in speed and capabilities. The time required by all the heuristics presented in this chapter would be greatly reduced by executing the code on today's faster PC's.

As illustrated in Figure 4-9, 68% of the runs required fewer than 4,000 evaluations.

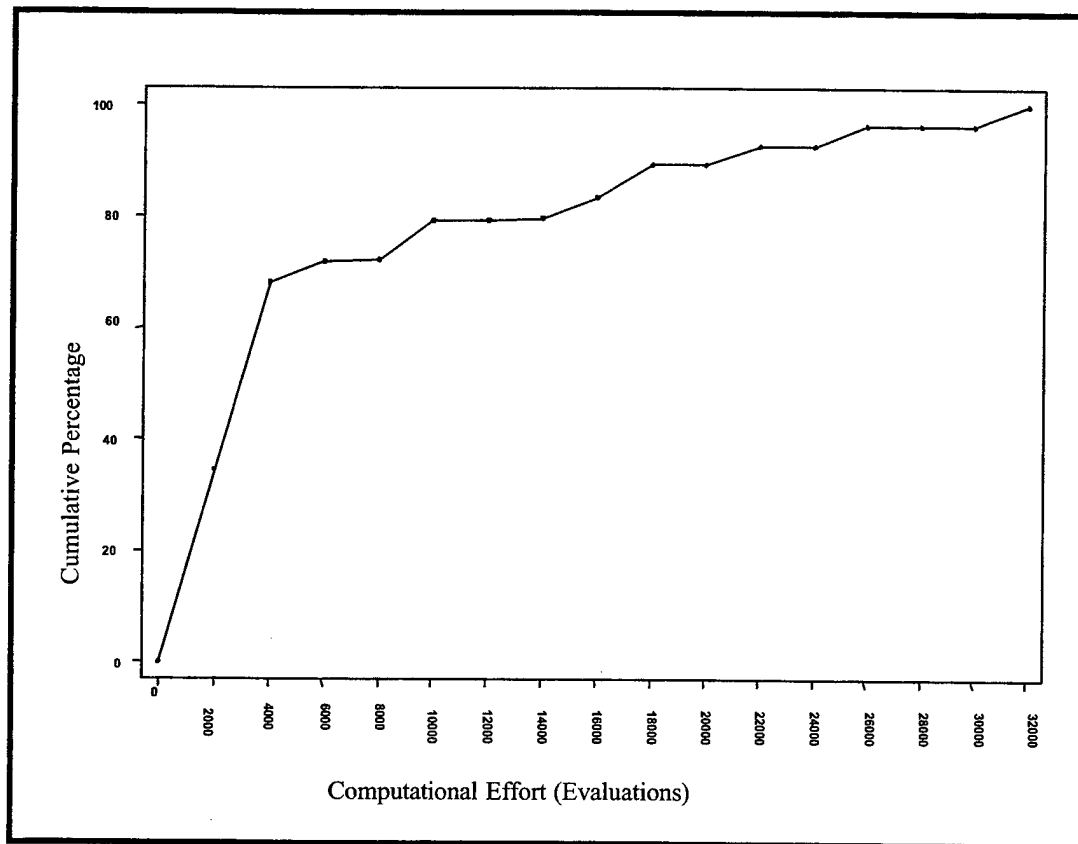


Figure 4-9. Performance vs. Evaluations

Functional evaluations are used as a measure of computational effort to compare different procedures running on different machines. All of the SA runs that required 20,000+ functional

evaluations are associated with the third DOE/RS problem (16/15). As expected, a completely saturated design will generate a high ratio of rejections to acceptances due to the unavailability of unassigned columns. The higher the ratio of rejections to acceptances, the more functional evaluations performed.

Overall, DESA performed well in solving the DOE/RS problems with an average solution value within 96% of the optimum value. The ICAGE heuristic outperformed it with an average solution value of 99.4% of the optimum value.

4.5 Summary

This chapter presented three heuristic methodologies for solving the DOE/RS problem. Column Assignment Via Examination (CAVE), Iterative Column Assignment Via Examination (ICAVE), and Design of Experiments using Simulated Anneling (DESA). CAVE and ICAGE are new heuristics that exploit the structure of the 2^n full factorial experimental designs. DESA is a standard heuristic applied for the first time to the DOE/RS problem.

All of the heuristics ran on desktop PCs and performed well in terms of speed and solution quality. CAVE provides a quick greedy heuristic that creates an excellent starting solution for other optimization methods if the solution CAVE produces is not optimal. ICAGE solutions were superior to DESA solutions, however ICAGE tended to use more evaluations than DESA. Considering the low cost of desktop PC computational time, ICAGE appears to be the superior heuristic for solving the DOE/RS problem. This result is not surprising since simulated annealing is a general-purpose heuristic technique and ICAGE was specifically designed to exploit the structure of the DOE/RS problem. The ability to solve large DOE/RS

problems using only the computational capabilities of a desktop PC gives an experimenter new choices in designing and running efficient experimental programs.

While these first steps are important, there is room for improvement. Alternative ICAVE stopping criteria could potentially improve the efficiency of the ICAVE heuristic. The DESA parameters could be adjusted based on the characteristics of the DOE/RS problem. The starting temperature could be adjusted based on the results of a CAVE solution. These improvements could be the focus of future research in the area of computer-aided experimental design.

Chapter Five compares the performance of all the solution techniques: the IP approach and the three heuristic approaches.

V. Conclusions

5.1 Summary

The research documented in this dissertation has developed four methodologies for computer-aided experimental design of two-level fractional factorial designs with requirement sets (DOE/RS): The 0-1 IP formulation, CAVE, ICAVE, and DESA. These methodologies extend the size of DOE/RS problems beyond that which can be solved by current manual techniques. Their effectiveness at solving heretofore intractably large DOE/RS problems demonstrates a significant contribution to DOE. The DOE/RS problem was proven to be NP-Complete.

The 0-1 IP formulation was developed as an algorithmic approach. The IP formulation provides a standard optimization methodology, which can guarantee that the solution is a global optimum for problems that fit within the dynamic memory resources available on the host computer. Due to the NP-Complete nature of the DOE/RS problem, the IP formulation can become computationally intractable as the size of the problem increases. However, when the IP finds a zero-valued solution it can terminate the branch-and-bound search.

Two heuristics, CAVE and ICAVE, are specifically designed to take advantage of the special structure of the two-level fractional factorial design. CAVE provides a quick feasible design that worked well on the 16-run designs and can also serve as a starting point for other solution techniques. ICAVE is an iterative variation of the CAVE heuristic. The ICAVE heuristic performance over all ten problems resulted in an average solution quality that was within 99% of the optimal solutions.

For comparison to a standard heuristic, a simulated annealing approach was used in the DESA heuristic. DESA performed consistently over a wide range of SA parameters and achieved an average of 96% of the optimum solution value for the ten DOE/RS problems.

5.2 Comparative Results

This section presents comparative data for the DOE/RS solution methodologies. It presents performance, computational effort, and the efficiency of these methodologies in solving DOE/RS problems.

5.2.1 Performance of the Solution Methodologies. Figure 5-1 presents the performance of the IP, DESA, and ICARE and CARE solution methodologies versus the ten DOE/RS problems solved in Chapters Three and Four. Notice that the scale is from 0.86 to 1 rather than from 0 to 1. The IP solution for problem 32/31 is less than optimal because the search was terminated due to computer limitations before a better solution was found. The heuristics methods all found better solutions to problem 32/31. Problems 16/15 and 64/63 also terminated due to computer limitations but their solution values were within 99% of the optimum value.

When averaged across the ten problems, the performance of each of the methodologies was within 93% (or better) of the optimal solution value.

5.2.2 Computational Effort of the Solution Methodologies. As seen in Figure 5-1, the computational effort required by the four solution methodologies varied dramatically. The 0-1 IP formulation performed millions of evaluations to establish that the solution found was optimal. The exceptions to this case occurred when a zero-valued solution was found. The IP methodology also had the most stringent computer requirements. In contrast, the heuristic methods performed well on what today are low-end PC's (386's and 486's).

The CARE heuristic required the least computational effort, but provided the weaker solutions. The ICARE computational effort increased as the problem size increased while the DESA computational effort was not a function of the problem size. Despite these limitations, all the heuristic methods enabled the rapid generation and evaluation of experimental designs.

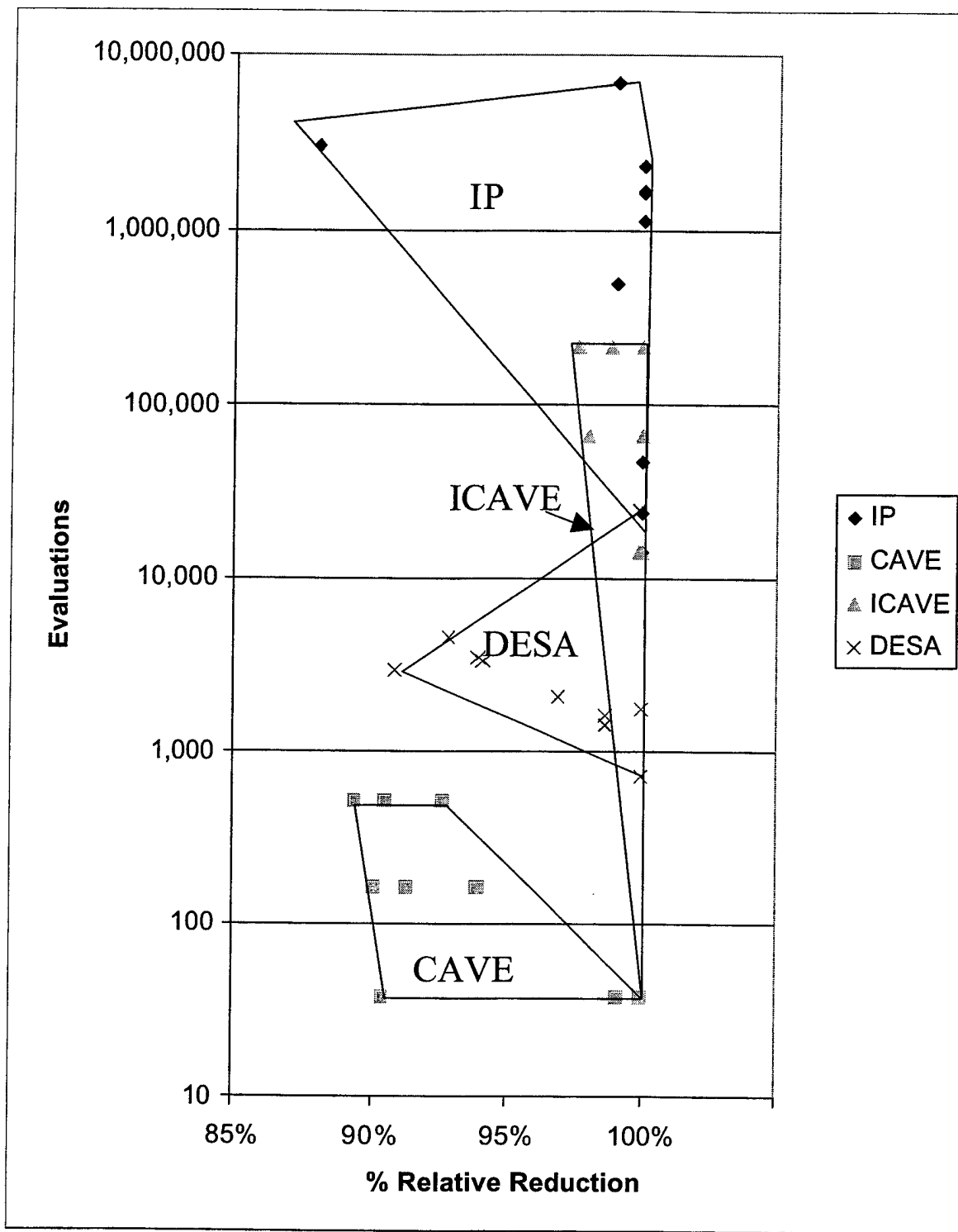


Figure 5-1. Mean Computational Effort versus Percentage of Relative Reduction for the Four Methodologies Under Study.

When choosing which of the solution methodologies to use to solve a DOE/RS problem, an experimenter should understand that each methodology has certain characteristics that could make any one of them the methodology of choice. In terms of computational effort, the methodologies can be ranked CAVE (fastest), DESA, ICAVE, 0-1 IP. ICAVE proved to be a highly effective solution methodology that used fewer computational resources than the IP approach required. The 0-1 IP is a standard methodology and can prove a solution is the global optimum solution.

5.2.3 Efficiencies Achieved by Solving DOE/RS Problems. The methodologies developed in this research allow an experimenter to solve a DOE/RS problem that involved a RS of 12 to 63 terms. An experimenter has the option of choosing a larger design that allows the estimation of all the two-factor interaction terms. Table 5-1 presents a comparison of the efficiency of utilizing a DOE/RS methodology over a resolution V design.

TABLE 5-1: COMPARISON OF DOE/RS DESIGN TO R_V DESIGN

# of First-Order Terms	# of Interaction Term in RS	Size of Potential DOE/RS Solution	Size of R_V Design
7	0-8 (9-24)	16 (32)	64
8	0-7 (8-23)	16 (32)	128
9	0-6 (7-22)	16 (32)	128
10	0-5 (6-21)	16 (32)	128
12	0-3 (4-19)	16 (32)	256
16	0-15 (16-31)	32 (64)	256
20	0-11 (12-27)	32 (64)	512

Utilizing the DOE/RS approach, an experimenter can reduce the number of experimental design points by a factor of 1/2 to 1/16 depending on the specifics of the DOE/RS problem. A reduction of this magnitude could represent tremendous savings in time, experimental resources, and money.

5.3 Recommendations for Future Research

Like all research, this investigation has limitations and boundaries that provide opportunities for future contributions. The opportunities for improving these methods, analyzing their performance, and developing new methods are numerous.

Some opportunities include:

- Investigate methods of identifying the appropriate weights for an experimenter to use when solving a DOE/RS problem. It may be possible to develop a means to translate risk or cost estimates into appropriate weights.
- Additional research could be done on the impact of different weights on the various optimization methodologies employed in this research. How would changing all the C_i 's to one change the efficiency and effectiveness of each of the optimization techniques? A weighting scheme of all ones would change the fitness landscape and therefore potentially affect the DESA heuristic (Hajek, 1988). An IP approach might also be affected since the convex hull it searches is likely to contain many more degenerate points (Bazaraa, *et.al.*, 1977, p. 164). The performances of the CAVE and ICAVE heuristics are probably not sensitive to changes in the weights.
- The most efficient number of iterations for the ICAVE heuristic is worthy of study. Repeated attempts with fewer iterations may be more effective. The most efficient number of iterations may be a function of the characteristics of the DOE/RS problem.
- CAVE appeared to be very effective at solving the 2^4 size design problems. The effectiveness of CAVE in solving designs of 16 runs or less is worthy of further study.

- Apply these newly developed methods to additional real world experiments. The application of scientific knowledge to aid decision-makers is fundamental to Operations Research.
- Develop computer-aided methods for experimental designs other than two-level fractional factorial designs.
- Investigate other computer-aided approaches for solving large DOE/RS problems such as Tabu search (Glover, 1990 or Glover and Laguna, 1997) and Genetic Algorithms (Tu, 1992 or Fogel, 1999) and compare with the methods presented here.
- Integrate these DOE/RS methods into an accessible, user-friendly software package. Perhaps these techniques could be incorporated into existing DOE software or as an add-on to a widely available product such as MS Excel.
- All of the solution methodologies could benefit from parallelization and optimization. Future research might attempt to establish more efficient code for single processor as well as multiple processor implementations.
- Refine the starting temperature value for DESA based on the lowest CAVE effectiveness values. In concert with this change, a slower cooling schedule would enhance performance for the same solution time.
- New metrics could be devised and examined to identify the quality of a two-level fractional factorial design. Two possible metrics suggested in Chapter Three include the number of main effects and the number of interaction effects that can be clearly estimated.

5.4 Conclusion

This research developed, evaluated, and compared four computer-based methods for quickly and easily designing and evaluating two-level fractional factorial experimental designs with requirement sets. These computer-based methods allow experimenters to use smaller, more effective experimental

designs without resorting to difficult to use manual methods. Smaller experimental designs mean fewer experimental runs resulting in savings of money, time, effort, and resources.

VI. Bibliography

Aarts, E. and Korst, J. *Simulated Annealing and Boltzmann Machines*. Tiptree Essex, Great Britain: John Wiley & Sons, (1989).

Anily, S. and Federgruen, A. "Simulated Annealing Methods with General Acceptance Probabilities," *Journal of Applied Probability*, Vol. 24, 657-667 (1987).

Barr, R.S., Golden, B.L., Kelly, J.P., Resende, M.G.C. and Stewart, W.R. "Designing and Reporting on Computational Experiments with Heuristic Methods," *Heuristics*, Vol. 1, No. 1 (1995).

Barzilai, J. and Golany, B. "Deriving Weights from Pairwise Comparisons Matrices: The Additive Case," *Operations Research Letters*, Vol. 9, 407-410 (1990).

Bazaraa, M.S., Jarvis, J.J., Sherali, H.D. *Linear Programming and Network Flows*. New York, John Wiley & Sons, 164, (1977)

Beale, E.M.L. and Tomlin, J. "Special Facilities in General Mathematical Programming System for Non-Convex Problems Using Ordered Sets of Variables", Proc. 5th IFORS Conference, J.R. Lawrence (ed.), New York, Wiley & Sons, (1970)

Bethke, A.D. *Genetic Algorithms as Function Optimizers*, Doctoral Dissertation, University of Michigan, Ann Arbor, Michigan, (1980).

Bisgaard, S. and Fuller, H.T. "Accommodating Interaction Effects in Two-Level Fractional Factorials: An Alternative to Linear Graphs," *Quality Engineering*, Vol. 7, No. 1, 71-87 (1995).

Bohachevsky, I.O., Johnson, M.E. and Stein, M.L. "Generalized Simulated Annealing for Function Optimization," *Technometrics*, Vol. 28, No. 3, 209-217 (1986).

Boese, K.D., Kahng, A.B. and Tsao, A.C.W. "Best-So-Far Vs. Where-You-Are: New Directions in Simulated Annealing For CAD," *Computer Science Department Technical Report*, University of California, Los Angeles, California, Dec. (1992) CSD-920050.

Bononi, E. and Lutton, J.L. "The N-City Traveling Salesman Problem: Statistical Mechanics and the Metropolis Algorithm," *SIAM Review*, Vol. 26, 551-568 (1984).

Box, G.E.P. and Draper, N.R. *Empirical Model-Building and Response Surfaces*. New York: John Wiley & Sons, 1-17 and 105-202, (1987).

Box, G.E.P., Hunter, W.G., and Hunter, J.S. *Statistics for Experimenters. An Introduction to Design, Data Analysis, and Model Building*. New York: John Wiley & Sons, (1978).

Cardoso, M.F., Salcedo, R.L. and de Azevedo, S.F. "Nonequilibrium Simulated Annealing: A Faster Approach to Combinatorial Minimization," *Industrial & Engineering Chemical Research*, Vol. 33, 1909-1918 (1994).

Cerny, V. "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm," *Journal of Optimization Theory and Applications*, Vol. 45, 41-51 (1985).

Clemen, R.T. *Making Hard Decisions: An Introduction to Decision Analysis*, Boston, PWS-Kent Publishing Co. (1996).

Collins, N.E., Eglese, R.W. and Golden, B.L. "Simulated Annealing - An Annotated Bibliography," *Working Paper Series MS/S 88-019*, University of Cambridge, UK, Oct. (1988).

Conoly, D. "General Purpose Simulated Annealing," *Journal of Optimization Research Society*, Vol. 43, No. 5, 495-505 (1992).

Daellenbach, H.G., George, J.A. and McNickle, D.C. *Introduction to Operations Research Techniques, 2ed.*, London: Allen and Bacon, Inc, 647, (1983).

Deutsch, C.V. *Annealing Techniques Applied to Reservoir Modeling and the Integration of Geological and Engineering (Well Test) Data*, Doctoral Dissertation, Department of Applied Earth Sciences, Stanford University, Stanford, California, (1992).

Fishburn, P.C. "Independence in Utility Theory with Whole Product Sets," *Operations Research*, Vol. 13, 28-43 (1965).

Fogel, D.B. *Evolutionary Computation: Towards a New Philosophy of Machine Intelligence*, New York, IEEE Press, (1999)

Forsythe, S.L. *Optimization of the Air Apportionment in a TAC THUNDER Scenario Using Response Surface Methodology*, MS Thesis, Air Force Institute of Technology, Wright Patterson Air Force Base, March (1994) (AFIT/GST/ENS/94M-02).

Forsythe, S.L., Auclair, P., Deckro, R. and Shedden, J. *The Use of Simulated Annealing to Select a Minimum Aberration 2^{k-p} Fractional Factorial Design When Certain Interactions Are Important*, Working Paper, Air Force Institute of Technology, (1997).

- Franklin, M.F. "Selecting Defining Contrasts and Confounded Effects in p^{n-m} Factorial Experiments," *Technometrics*, Vol. 27, No. 2, 165-172 (1985).
- Fries, A. and Hunter, W.G. "Minimum Aberration Designs," *Technometrics*, Vol. 22, No. 4, 601-608 (1980).
- Garey, M.R. and Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco: W. H. Freeman & Co., (1979).
- Glover, F. "Tabu Search: A Tutorial," *Interfaces*, Vol. 20, Jul/Aug, 74-94 (1990).
- Glover, F. and Laguna, M. *Tabu Search*, Boston, Kluwer Academic Publishers, (1997).
- Greenfield, A.A. "Selection of Defining Contrasts in Two-Level Experiments," *Applied Statistics*, Vol. 25, 64-67 (1976).
- Haines, L.M. "The Application of the Annealing Algorithm to the Construction of Exact Optimal Designs for Linear-Regression Models," *Technometrics*, Vol. 29, No. 4, 439-447 (1987).
- Hajek, B. "Cooling Schedules for Optimal Annealing," *Mathematics of Operations Research*, Vol. 13, No. 2, 311-329 (1988).
- Harris, C.M., Hoffman, K.L. and Yarrow, L.A. "Using Integer Programming Techniques for the Solution of an Experimental Design Problem," *Annals of Operations Research*, Vol. 58, 243-260 (1995).
- Heine, G.W. *Smart Simulated Annealing*, Doctoral Dissertation, Department of Applied Mathematics, University of Colorado at Denver, Denver, Colorado, (1994).
- Hicks, C. R. *Fundamental Concepts in the Design of Experiments*, 3rd. ed., Holt, Rinehart and Winston, New York, NY, (1982).
- Hockman, K.K. and Jenkins, M.W. "Design of Experiments: Neglected Key to Competitive R&D," *Industrial Engineering*, Vol. 26, No. 2, 50-51 (1994).
- Hockman, K.K. and Jenkins, M.W. "Design of Experiments: Key to Competitive R&D," *Quality*, 47-50 (June 1994).
- Ingber, L. "Simulated Annealing: Practice Versus Theory," *Mathematical Computer Modeling*, Vol. 18, 29-58 (1993).

Johnson, D.S, Aragon, C.R., McGeoch, L.A. and Schevon, C. "Optimization by Simulated Annealing: An Experimental Evaluation; Part II, Graph Coloring and Number Partitioning," *Operations Research*, Vol. 39, No. 3, 378-406 (1991).

Keeney, R.L. "Utility Independence and Preferences for Multiattributed Consequences," *Operations Research*, Vol. 19, 875-893 (1971).

Keeney, R.L. and Raiffa, H. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York: John Wiley & Sons, Inc., (1976).

Keeney, R.L. *Value-Focused Thinking: A Path to Creative Decisionmaking*. London: Harvard University Press, 133-139, (1992).

Kirkpatrick, S.S, Gelatt Jr., C.D. and Vecchi, M. P. "Optimization by Simulated Annealing," *Science*, Vol. 220, No. 4598, 671-679 (1983).

Lam, J.K.C. *An Efficient Simulated Annealing Schedule*, Doctoral Dissertation, Yale University, New Haven, Connecticut, (1988).

Lee, S.H. *Efficient Design and Optimization of Robust Parameter Experiments*, Doctoral Dissertation, Department of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, Georgia, (1991).

Mandu, C.N. and Kuei, C. "Group Screening and Taguchi Design in the Optimization of Multi-Echelon Maintenance Float Simulation Metamodels," *Computer and Operations Research*, Vol. 19, No. 2, 95-105 (1992).

Mee, R. W. and Peralta M. "Semifolding 2^{k-p} Designs," *Technometrics*, Vol. 42, No. 2, 122-134 (2000).

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A. and Teller, E "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, Vol. 21, 1087-1092 (1953).

Mongomery, D. C., Myers, R. H. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. New York. Wiley & Sons, (1995).

Mongomery, D. C. *Design and Analysis of Experiments*. New York. Wiley & Sons, (1999).

Nachtsheim, C.J. "Tools for Computer-Aided Design of Experiments," *Journal of Quality Technology*, Vol. 19, No. 3, 132-160 (1987).

- Patel, A.N. *Heuristic Synthesis in Design of Noncontinuous Plants*, Doctoral Dissertation, Department of Chemical Engineering, Northwestern University, Evanston, Illinois, (1993).
- Sangiovanni-Vincentelli, A. "Editor's Foreward," *Algorithmica*, Vol. 6, No. 3, 295-301 (1991).
- Schmidt, S.R. and Launsby, R.G. *Understanding Industrial Designed Experiments*. Colorado Springs, Colorado: Air Academy Press, 1-20, (1991).
- Shannon, R.E. *Systems Simulation - the Art and Science*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 144-179, (1975).
- Steuer, R. E. *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley, New York, (1986).
- Stewart, T.J. "Inferring Preferences in Multiple Criteria Decision Analysis Using a Logistic Regression Model," *Management Science*, Vol. 30, No. 9, 1067-1077 (1984).
- Stewart, T.J. "Pruning of Decision Alternatives in Multiple Criteria Decision Making, Based on the UTA Method for Estimating Utilities," *European Journal of Operations Research*, Vol. 28, 79-88 (1987).
- Taguchi, G. and Wu, Y. *Introduction to Off-Line Quality Control*. Central Japan Quality Control Association, Nagoya, Japan, (1980).
- Taguchi, G. and Konishi, S. *Orthogonal Arrays and Linear Graphs: Tools for Quality Engineering*. American Supplier Institute, (1987).
- Tu, J.Z. *Genetic Algorithms in Machine Learning and Optimization*, Doctoral Dissertation, University of Cincinnati, Cincinnati, Ohio, (1992).
- Wilson, J.M. "Generating Cuts in Integer Programming with Families of Special Ordered Sets," *European Journal of Operations Research*, Vol. 46, 101-108 (1990).
- Winston, W.L. *Operations Research: Applications and Algorithms*. Belmont, California: Duxbury Press, 772-774, (1994).
- Woodruff, D.L. "Simulated Annealing and Tabu Search: Lessons from a Line Search," *Computer and Operations Research*, Vol. 21, No. 8, 823-839 (1994).
- Wu, C.F.J. and Chen, Y. "A Graph-Aided Method for Planning Two-Level Experiments When Certain Interactions Are Important," *Technometrics*, Vol. 34, No. 4, 162-175 (1992).

VII. Appendices

A. Experimental Results from DESA Experiments

The following table provides the detailed results of the SA experiment. The headings are as follows:

- Run #: Each of the 27 design points was run ten times. Runs 1-10 were conducted with settings from the first design point, 11-20 with the second, and so on.
- ExDes Pt: This column identifies which of the 27 design points the run corresponds to.
- S: This column identifies the size of the design matrix. -1 = 16, 0 = 32, +1 = 64.
- TL: Temperature Length controls the number of perturbations which occur at each temperature level. When the number of acceptances equals TL or when a total of $20*TL$ perturbations occur the temperature is decreased. -1 = 5, 0 = 15, 1 = 20
- CM: The cooling multiplier controls the magnitude of each temperature decrease. $T_{i+1} = (cm)T_i$. -1 = .90, 0 = .93, 1 = .96
- ST: The stopping temperature causes the SA procedures to terminate when the temperature is less than ST . -1 = 0.1, 0 = 10, 1 = 20
- %UC: The percentage of unassigned columns (%UC) is approximately (10%, 20%, and 0%): $1 - [(\# \text{ terms in RS})/(\# \text{ of columns in design matrix})]$. -1 = 0, 0 = 10%, 1 = 20%
- DESA Soltn: This entry is the best solution found by the SA search.
- Evaluations: This entry is the number of evaluations utilized in the SA search.
- Best Kn: This entry is the best known solution to the particular baseline DOE/RS problem associated with the design point.

- Total Ci's: This entry is the sum of all the Ci values for the particular baseline DOE/RS problem associated with the design point.
- % RR: The metric "percentage of relative reduction" is calculated as shown in equation A-1. The variables with the "*" are those associated with the best known solution; the variables with the "/" are the variables associated with the solution the metric is measuring.

$$1 - \frac{\left(\sum_{i=1}^N C_i (1-x_i^*) + C_{\min ab} (1-y^*) \right) - \left(\sum_{i=1}^N C_i (1-x_i') + C_{\min ab} (1-y') \right)}{\left(\sum_{i=1}^N C_i (1-x_i^*) + C_{\min ab} (1-y^*) \right)} \quad (\text{A-1})$$

TABLE A-1: RESULTS OF DESA EXPERIMENTS

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
1	1	-1	-1	-1	-1	-1	51	31660	51	820	1.000
2	1	-1	-1	-1	-1	-1	51	31756	51	820	1.000
3	1	-1	-1	-1	-1	-1	51	31660	51	820	1.000
4	1	-1	-1	-1	-1	-1	51	31743	51	820	1.000
5	1	-1	-1	-1	-1	-1	51	31759	51	820	1.000
6	1	-1	-1	-1	-1	-1	51	31682	51	820	1.000
7	1	-1	-1	-1	-1	-1	51	31474	51	820	1.000
8	1	-1	-1	-1	-1	-1	51	31940	51	820	1.000
9	1	-1	-1	-1	-1	-1	51	31784	51	820	1.000
10	1	-1	-1	-1	-1	-1	51	31659	51	820	1.000
11	2	-1	-1	0	0	0	27	1464	17	791	0.987
12	2	-1	-1	0	0	0	27	1316	17	791	0.987
13	2	-1	-1	0	0	0	28	1168	17	791	0.986
14	2	-1	-1	0	0	0	27	1341	17	791	0.987
15	2	-1	-1	0	0	0	27	1374	17	791	0.987
16	2	-1	-1	0	0	0	30	1417	17	791	0.983
17	2	-1	-1	0	0	0	28	1436	17	791	0.986
18	2	-1	-1	0	0	0	27	1378	17	791	0.987
19	2	-1	-1	0	0	0	27	1351	17	791	0.987
20	2	-1	-1	0	0	0	27	1075	17	791	0.987
21	3	-1	-1	1	1	1	27	1406	17	778	0.987
22	3	-1	-1	1	1	1	27	1335	17	778	0.987
23	3	-1	-1	1	1	1	27	1471	17	778	0.987
24	3	-1	-1	1	1	1	27	1534	17	778	0.987
25	3	-1	-1	1	1	1	27	1385	17	778	0.987
26	3	-1	-1	1	1	1	27	1676	17	778	0.987
27	3	-1	-1	1	1	1	27	1451	17	778	0.987
28	3	-1	-1	1	1	1	27	1450	17	778	0.987
29	3	-1	-1	1	1	1	27	1387	17	778	0.987

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
30	3	-1	-1	1	1	1	27	1415	17	778	0.987
31	4	-1	0	-1	0	1	27	1607	17	778	0.987
32	4	-1	0	-1	0	1	27	1304	17	778	0.987
33	4	-1	0	-1	0	1	27	1485	17	778	0.987
34	4	-1	0	-1	0	1	27	1652	17	778	0.987
35	4	-1	0	-1	0	1	27	1768	17	778	0.987
36	4	-1	0	-1	0	1	27	1676	17	778	0.987
37	4	-1	0	-1	0	1	27	1610	17	778	0.987
38	4	-1	0	-1	0	1	27	1636	17	778	0.987
39	4	-1	0	-1	0	1	27	1288	17	778	0.987
40	4	-1	0	-1	0	1	27	1615	17	778	0.987
41	5	-1	0	0	1	-1	53	1982	51	820	0.997
42	5	-1	0	0	1	-1	53	1833	51	820	0.997
43	5	-1	0	0	1	-1	51	1686	51	820	1.000
44	5	-1	0	0	1	-1	53	1854	51	820	0.997
45	5	-1	0	0	1	-1	53	1760	51	820	0.997
46	5	-1	0	0	1	-1	51	1911	51	820	1.000
47	5	-1	0	0	1	-1	51	1760	51	820	1.000
48	5	-1	0	0	1	-1	51	1760	51	820	1.000
49	5	-1	0	0	1	-1	53	1885	51	820	0.997
50	5	-1	0	0	1	-1	53	2006	51	820	0.997
51	6	-1	0	1	-1	0	27	21630	17	791	0.987
52	6	-1	0	1	-1	0	27	21849	17	791	0.987
53	6	-1	0	1	-1	0	27	21899	17	791	0.987
54	6	-1	0	1	-1	0	27	21725	17	791	0.987
55	6	-1	0	1	-1	0	27	21871	17	791	0.987
56	6	-1	0	1	-1	0	27	21796	17	791	0.987
57	6	-1	0	1	-1	0	27	21791	17	791	0.987
58	6	-1	0	1	-1	0	27	21900	17	791	0.987
59	6	-1	0	1	-1	0	27	21881	17	791	0.987
60	6	-1	0	1	-1	0	27	16268	17	791	0.987
61	7	-1	1	-1	1	0	27	1464	17	791	0.987
62	7	-1	1	-1	1	0	27	1316	17	791	0.987
63	7	-1	1	-1	1	0	28	1168	17	791	0.986
64	7	-1	1	-1	1	0	27	1341	17	791	0.987
65	7	-1	1	-1	1	0	27	1374	17	791	0.987
66	7	-1	1	-1	1	0	30	1417	17	791	0.983
67	7	-1	1	-1	1	0	28	1436	17	791	0.986
68	7	-1	1	-1	1	0	27	1378	17	791	0.987
69	7	-1	1	-1	1	0	27	1351	17	791	0.987
70	7	-1	1	-1	1	0	27	1075	17	791	0.987
71	8	-1	1	0	-1	1	27	3711	17	778	0.987
72	8	-1	1	0	-1	1	27	3596	17	778	0.987
73	8	-1	1	0	-1	1	27	3229	17	778	0.987
74	8	-1	1	0	-1	1	27	3569	17	778	0.987
75	8	-1	1	0	-1	1	27	3563	17	778	0.987
76	8	-1	1	0	-1	1	27	3533	17	778	0.987

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
77	8	-1	1	0	-1	1	27	3248	17	778	0.987
78	8	-1	1	0	-1	1	27	3232	17	778	0.987
79	8	-1	1	0	-1	1	27	3718	17	778	0.987
80	8	-1	1	0	-1	1	27	3863	17	778	0.987
81	9	-1	1	1	0	-1	51	24834	51	820	1.000
82	9	-1	1	1	0	-1	51	24481	51	820	1.000
83	9	-1	1	1	0	-1	51	24288	51	820	1.000
84	9	-1	1	1	0	-1	51	24693	51	820	1.000
85	9	-1	1	1	0	-1	51	24327	51	820	1.000
86	9	-1	1	1	0	-1	51	24556	51	820	1.000
87	9	-1	1	1	0	-1	51	24574	51	820	1.000
88	9	-1	1	1	0	-1	51	24532	51	820	1.000
89	9	-1	1	1	0	-1	51	24684	51	820	1.000
90	9	-1	1	1	0	-1	51	24647	51	820	1.000
91	10	0	-1	-1	-1	-1	232	3248	0	1696	0.863
92	10	0	-1	-1	-1	-1	179	3206	0	1696	0.894
93	10	0	-1	-1	-1	-1	189	3457	0	1696	0.889
94	10	0	-1	-1	-1	-1	137	3325	0	1696	0.919
95	10	0	-1	-1	-1	-1	261	3328	0	1696	0.846
96	10	0	-1	-1	-1	-1	200	3331	0	1696	0.882
97	10	0	-1	-1	-1	-1	154	3290	0	1696	0.909
98	10	0	-1	-1	-1	-1	74	3329	0	1696	0.956
99	10	0	-1	-1	-1	-1	53	3226	0	1696	0.969
100	10	0	-1	-1	-1	-1	118	3328	0	1696	0.930
101	11	0	-1	0	0	0	78	1606	0	1606	0.951
102	11	0	-1	0	0	0	132	1579	0	1606	0.918
103	11	0	-1	0	0	0	125	1567	0	1606	0.922
104	11	0	-1	0	0	0	114	1567	0	1606	0.929
105	11	0	-1	0	0	0	130	1531	0	1606	0.919
106	11	0	-1	0	0	0	34	1659	0	1606	0.979
107	11	0	-1	0	0	0	34	1571	0	1606	0.9789
108	11	0	-1	0	0	0	53	1672	0	1606	0.967
109	11	0	-1	0	0	0	62	1627	0	1606	0.961
110	11	0	-1	0	0	0	89	1569	0	1606	0.945
111	12	0	-1	1	1	1	32	1885	0	1525	0.979
112	12	0	-1	1	1	1	37	1998	0	1525	0.976
113	12	0	-1	1	1	1	48	1929	0	1525	0.969
114	12	0	-1	1	1	1	0	544	0	1525	1.000
115	12	0	-1	1	1	1	0	950	0	1525	1.000
116	12	0	-1	1	1	1	38	1774	0	1525	0.975
117	12	0	-1	1	1	1	35	1702	0	1525	0.977
118	12	0	-1	1	1	1	0	1597	0	1525	1.000
119	12	0	-1	1	1	1	29	1932	0	1525	0.981
120	12	0	-1	1	1	1	0	1637	0	1525	1.000
121	13	0	0	-1	0	1	44	2118	0	1525	0.971
122	13	0	0	-1	0	1	0	815	0	1525	1.000
123	13	0	0	-1	0	1	0	1447	0	1525	1.000

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
124	13	0	0	-1	0	1	0	793	0	1525	1.000
125	13	0	0	-1	0	1	0	1211	0	1525	1.000
126	13	0	0	-1	0	1	38	1960	0	1525	0.975
127	13	0	0	-1	0	1	0	894	0	1525	1.000
128	13	0	0	-1	0	1	35	2102	0	1525	0.977
129	13	0	0	-1	0	1	0	1264	0	1525	1.000
130	13	0	0	-1	0	1	35	2134	0	1525	0.977
131	14	0	0	0	1	-1	156	2187	0	1696	0.908
132	14	0	0	0	1	-1	177	2218	0	1696	0.896
133	14	0	0	0	1	-1	129	2261	0	1696	0.924
134	14	0	0	0	1	-1	171	2235	0	1696	0.899
135	14	0	0	0	1	-1	160	2315	0	1696	0.906
136	14	0	0	0	1	-1	127	2167	0	1696	0.925
137	14	0	0	0	1	-1	163	2353	0	1696	0.904
138	14	0	0	0	1	-1	151	2047	0	1696	0.911
139	14	0	0	0	1	-1	168	2225	0	1696	0.901
140	14	0	0	0	1	-1	230	2240	35	1696	0.883
141	15	0	0	1	-1	0	0	3690	0	1606	1.000
142	15	0	0	1	-1	0	33	17100	0	1606	0.979
143	15	0	0	1	-1	0	48	17046	0	1606	0.970
144	15	0	0	1	-1	0	35	16981	0	1606	0.978
145	15	0	0	1	-1	0	40	17239	0	1606	0.975
146	15	0	0	1	-1	0	35	16827	0	1606	0.978
147	15	0	0	1	-1	0	77	17168	0	1606	0.952
148	15	0	0	1	-1	0	88	16827	0	1606	0.945
149	15	0	0	1	-1	0	35	17056	0	1606	0.978
150	15	0	0	1	-1	0	36	16625	0	1606	0.978
151	16	0	1	-1	1	0	71	2022	0	1606	0.956
152	16	0	1	-1	1	0	108	2110	0	1606	0.933
153	16	0	1	-1	1	0	33	2457	0	1606	0.979
154	16	0	1	-1	1	0	0	1622	0	1606	1.000
155	16	0	1	-1	1	0	76	2251	0	1606	0.953
156	16	0	1	-1	1	0	43	2052	0	1606	0.973
157	16	0	1	-1	1	0	0	1741	0	1606	1.000
158	16	0	1	-1	1	0	74	2296	0	1606	0.954
159	16	0	1	-1	1	0	47	2275	0	1606	0.971
160	16	0	1	-1	1	0	87	2010	0	1606	0.946
161	17	0	1	0	-1	1	0	1576	0	1525	1.000
162	17	0	1	0	-1	1	0	2020	0	1525	1.000
163	17	0	1	0	-1	1	0	2772	0	1525	1.000
164	17	0	1	0	-1	1	0	1090	0	1525	1.000
165	17	0	1	0	-1	1	0	2772	0	1525	1.000
166	17	0	1	0	-1	1	30	13896	0	1525	0.980
167	17	0	1	0	-1	1	0	1719	0	1525	1.000
168	17	0	1	0	-1	1	0	2206	0	1525	1.000
169	17	0	1	0	-1	1	0	1430	0	1525	1.000
170	17	0	1	0	-1	1	0	2800	0	1525	1.000

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
171	18	0	1	1	0	-1	89	8168	0	1696	0.948
172	18	0	1	1	0	-1	102	8219	0	1696	0.940
173	18	0	1	1	0	-1	62	8575	0	1696	0.963
174	18	0	1	1	0	-1	103	8358	0	1696	0.939
175	18	0	1	1	0	-1	99	8558	0	1696	0.942
176	18	0	1	1	0	-1	123	8028	0	1696	0.927
177	18	0	1	1	0	-1	35	1702	0	1696	0.979
178	18	0	1	1	0	-1	119	8638	0	1696	0.930
179	18	0	1	1	0	-1	101	8170	0	1696	0.940
180	18	0	1	1	0	-1	83	8358	0	1696	0.951
181	19	1	-1	-1	-1	-1	775	3409	500	3716	0.914
182	19	1	-1	-1	-1	-1	590	3449	500	3716	0.972
183	19	1	-1	-1	-1	-1	749	3519	500	3716	0.923
184	19	1	-1	-1	-1	-1	606	3492	500	3716	0.97
185	19	1	-1	-1	-1	-1	683	3546	500	3716	0.943
186	19	1	-1	-1	-1	-1	661	3409	500	3716	0.950
187	19	1	-1	-1	-1	-1	851	3495	500	3716	0.891
188	19	1	-1	-1	-1	-1	758	3499	500	3716	0.920
189	19	1	-1	-1	-1	-1	726	3479	500	3716	0.930
190	19	1	-1	-1	-1	-1	656	3444	500	3716	0.951
191	20	1	-1	0	0	0	344	1752	135	3353	0.935
192	20	1	-1	0	0	0	399	1749	135	3353	0.918
193	20	1	-1	0	0	0	602	1835	135	3353	0.855
194	20	1	-1	0	0	0	552	1667	135	3353	0.870
195	20	1	-1	0	0	0	337	1854	135	3353	0.937
196	20	1	-1	0	0	0	718	1792	135	3353	0.819
197	20	1	-1	0	0	0	386	1848	135	3353	0.922
198	20	1	-1	0	0	0	367	1706	135	3353	0.928
199	20	1	-1	0	0	0	479	1796	135	3353	0.893
200	20	1	-1	0	0	0	347	1759	135	3353	0.934
201	21	1	-1	1	1	1	393	2130	0	3026	0.870
202	21	1	-1	1	1	1	299	2222	0	3026	0.901
203	21	1	-1	1	1	1	171	2215	0	3026	0.943
204	21	1	-1	1	1	1	287	2014	0	3026	0.905
205	21	1	-1	1	1	1	263	2326	0	3026	0.913
206	21	1	-1	1	1	1	305	2067	0	3026	0.899
207	21	1	-1	1	1	1	174	2120	0	3026	0.942
208	21	1	-1	1	1	1	219	2269	0	3026	0.928
209	21	1	-1	1	1	1	132	2197	0	3026	0.956
210	21	1	-1	1	1	1	249	2262	0	3026	0.918
211	22	1	0	-1	0	1	194	4578	0	3026	0.936
212	22	1	0	-1	0	1	332	4397	0	3026	0.890
213	22	1	0	-1	0	1	212	4544	0	3026	0.930
214	22	1	0	-1	0	1	299	4446	0	3026	0.901
215	22	1	0	-1	0	1	156	4580	0	3026	0.948
216	22	1	0	-1	0	1	204	4660	0	3026	0.933
217	22	1	0	-1	0	1	254	4480	0	3026	0.916

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
218	22	1	0	-1	0	1	120	4507	0	3026	0.960
219	22	1	0	-1	0	1	224	4539	0	3026	0.926
220	22	1	0	-1	0	1	308	4568	0	3026	0.898
221	23	1	0	0	1	-1	707	2558	500	3716	0.936
222	23	1	0	0	1	-1	797	2830	500	3716	0.908
223	23	1	0	0	1	-1	752	2742	500	3716	0.922
224	23	1	0	0	1	-1	811	2540	500	3716	0.903
225	23	1	0	0	1	-1	845	2655	500	3716	0.893
226	23	1	0	0	1	-1	726	2855	500	3716	0.930
227	23	1	0	0	1	-1	799	2708	500	3716	0.907
228	23	1	0	0	1	-1	770	2574	500	3716	0.916
229	23	1	0	0	1	-1	720	2569	500	3716	0.932
230	23	1	0	0	1	-1	595	7667	500	3716	0.970
231	24	1	0	1	-1	0	316	17460	135	3353	0.944
232	24	1	0	1	-1	0	443	17570	135	3353	0.904
233	24	1	0	1	-1	0	399	17820	135	3353	0.918
234	24	1	0	1	-1	0	427	17769	135	3353	0.909
235	24	1	0	1	-1	0	264	17700	135	3353	0.960
236	24	1	0	1	-1	0	391	17631	135	3353	0.920
237	24	1	0	1	-1	0	401	2746	135	3353	0.917
238	24	1	0	1	-1	0	540	2683	135	3353	0.874
239	24	1	0	1	-1	0	456	2573	135	3353	0.900
240	24	1	0	1	-1	0	407	2678	135	3353	0.915
241	25	1	1	-1	1	0	465	2487	135	3353	0.897
242	25	1	1	-1	1	0	531	2628	135	3353	0.877
243	25	1	1	-1	1	0	486	2567	135	3353	0.891
244	25	1	1	-1	1	0	431	2583	135	3353	0.908
245	25	1	1	-1	1	0	538	2641	135	3353	0.875
246	25	1	1	-1	1	0	440	2606	135	3353	0.905
247	25	1	1	-1	1	0	401	2746	135	3353	0.917
248	25	1	1	-1	1	0	540	2683	135	3353	0.874
249	25	1	1	-1	1	0	456	2573	135	3353	0.900
250	25	1	1	-1	1	0	407	2678	135	3353	0.915
251	26	1	1	0	-1	1	283	14828	0	3026	0.906
252	26	1	1	0	-1	1	194	14683	0	3026	0.936
253	26	1	1	0	-1	1	172	14862	0	3026	0.943
254	26	1	1	0	-1	1	212	14664	0	3026	0.930
255	26	1	1	0	-1	1	129	14491	0	3026	0.957
256	26	1	1	0	-1	1	202	14660	0	3026	0.933
257	26	1	1	0	-1	1	261	14746	0	3026	0.914
258	26	1	1	0	-1	1	222	14701	0	3026	0.927
259	26	1	1	0	-1	1	111	15013	0	3026	0.963
260	26	1	1	0	-1	1	187	14686	0	3026	0.938
261	27	1	1	1	0	-1	609	9164	500	3716	0.966
262	27	1	1	1	0	-1	568	9562	500	3716	0.979
263	27	1	1	1	0	-1	573	9715	500	3716	0.977
264	27	1	1	1	0	-1	688	9540	500	3716	0.942

Run #	ExDes Pt	S	TL	CM	ST	%UC	DESA Soltn	Evaluations	Best Known	Total Ci's	% RR
265	27	1	1	1	0	-1	544	8806	500	3716	0.986
266	27	1	1	1	0	-1	500	9532	500	3716	1.000
267	27	1	1	1	0	-1	658	9490	500	3716	0.951
268	27	1	1	1	0	-1	792	9387	500	3716	0.909
269	27	1	1	1	0	-1	662	9385	500	3716	0.950
270	27	1	1	1	0	-1	625	9253	500	3716	0.961

B. Experimental Results from ICAVE Heuristic Experiments

This section provides additional information on the ICAVE data. The ICAVE methodology consisted of a maximum of 100 iterations of the CAVE heuristic. The search frequently terminated early when a zero-valued solution was found before the 100th iteration. In seven of the experimental designs, no variation in performance was observed. These results are presented in Table B-1.

TABLE B-1. ICAVE RESULTS FOR RUNS WITH ZERO DEVIATIONS

ID	Cost	Number of repetitions
16/11	0	5
16/12	17	5
16/13	18	5
16/15	41	5
32/25	0.0	10
32/28	0.0	5
64/51	0.0	5

The 32/25 experiment ran so quickly that twice the usual number of runs was completed in the same amount of time used for the other experiments.

The results of the remaining runs are presented in Table B-2.

TABLE B-2: ICAVE RESULTS FOR RUNS WITH POSITIVE DEVIATIONS

ID	Costs
32/31	50, 29, 27, 0, 66, 59, 37, 0
64/57	135, 211
64/63	550, 606

C. Bisgaard and Fuller Example Problem

Bisgaard and Fuller presented an example DOE/S problem which had eight factors (A, B, C, D, E, F, G, H) and three interaction terms (AB, AD, and BD). These terms together represent a requirement set of 11 terms. See Appendix H for details on the various notations used to represent columns of the design matrix. Identified as part of this problem was the design generator for a maximum resolution - minimum aberration design (5=234, 6 = 134, 7 = 123, and 8 = 124). Notice that column "234" (i.e BCD and (1,1,1,0)) is the same as column number 14 (i.e. 8+4+2+0). Likewise 134 (i.e. ACD or (1,1,0,1)) = column 13, 123 (i.e. ABC or (0,1,1,1)) = column 7, and 124 (i.e. ABD or (1,0,1,1)) = column 11.

This section presents the IP formulation for solving this problem. Subscripts were avoided in the code that produced the IP formulation to simplify writing the formulation. In the formulations presented in the Appendix, x_i is written Xi and x_{ij} is written XiCj meaning the i^{th} term in the RS is assigned to the j^{th} column. The code produces the DOE/RS problem in CPLEX's "LP Format". This format is an algebraic representation of the problem similar to how the problem might appear in an OR text.

In the Bisgaard-Fuller problem, X1 through X8 represent (A-H). X9, X10, and X11 represent AB, AD, and BD respectively. Y = 0 if all the first order terms are assigned to the columns associated with the specified minimum aberration design; otherwise, Y =1. C1 through C15 represent the 15 columns of the experimental design to which variables can be assigned.

The objective function will equal zero if none of the terms are confounded with any other term and if Y = 0, thereby indicating a minimum aberration design. Bisgaard and Fuller did not explicitly assign weights to the terms. They did state that the interaction terms were

presented in order of preference. In this example problem, the weights are irrelevant because a zero value solution exists. Changing the weights would change the convex hull of the LP solution space. The impact of different weighting schemes on the LP solution methodology remains a topic for future research.

Minimize

Obj: $100 X_1 + 100 X_2 + 100 X_3 + 100 X_4 + 100 X_5 + 100 X_6 + 100 X_7 + 100 X_8 + 10 X_9 + 9 X_{10} + 8 X_{11} + 500 Y$

Subject to

The assignment constraints force each variable to be assigned to one and only one column. For example, if variable A is assigned to column 1, then $X_{1C1} = 1$. Therefore X_{1C2} through X_{1C15} must equal zero. Since the RS has 11 terms, 11 constraints are needed: one for each term in the RS.

c 1 : $X_{1C1} + X_{1C2} + X_{1C3} + X_{1C4} + X_{1C5} + X_{1C6} + X_{1C7} + X_{1C8} + X_{1C9} + X_{1C10} + X_{1C11} + X_{1C12} + X_{1C13} + X_{1C14} + X_{1C15} = 1$
c 2 : $X_{2C1} + X_{2C2} + X_{2C3} + X_{2C4} + X_{2C5} + X_{2C6} + X_{2C7} + X_{2C8} + X_{2C9} + X_{2C10} + X_{2C11} + X_{2C12} + X_{2C13} + X_{2C14} + X_{2C15} = 1$
c 3 : $X_{3C1} + X_{3C2} + X_{3C3} + X_{3C4} + X_{3C5} + X_{3C6} + X_{3C7} + X_{3C8} + X_{3C9} + X_{3C10} + X_{3C11} + X_{3C12} + X_{3C13} + X_{3C14} + X_{3C15} = 1$
c 4 : $X_{4C1} + X_{4C2} + X_{4C3} + X_{4C4} + X_{4C5} + X_{4C6} + X_{4C7} + X_{4C8} + X_{4C9} + X_{4C10} + X_{4C11} + X_{4C12} + X_{4C13} + X_{4C14} + X_{4C15} = 1$
c 5 : $X_{5C1} + X_{5C2} + X_{5C3} + X_{5C4} + X_{5C5} + X_{5C6} + X_{5C7} + X_{5C8} + X_{5C9} + X_{5C10} + X_{5C11} + X_{5C12} + X_{5C13} + X_{5C14} + X_{5C15} = 1$
c 6 : $X_{6C1} + X_{6C2} + X_{6C3} + X_{6C4} + X_{6C5} + X_{6C6} + X_{6C7} + X_{6C8} + X_{6C9} + X_{6C10} + X_{6C11} + X_{6C12} + X_{6C13} + X_{6C14} + X_{6C15} = 1$
c 7 : $X_{7C1} + X_{7C2} + X_{7C3} + X_{7C4} + X_{7C5} + X_{7C6} + X_{7C7} + X_{7C8} + X_{7C9} + X_{7C10} + X_{7C11} + X_{7C12} + X_{7C13} + X_{7C14} + X_{7C15} = 1$
c 8 : $X_{8C1} + X_{8C2} + X_{8C3} + X_{8C4} + X_{8C5} + X_{8C6} + X_{8C7} + X_{8C8} + X_{8C9} + X_{8C10} + X_{8C11} + X_{8C12} + X_{8C13} + X_{8C14} + X_{8C15} = 1$
c 9 : $X_{9C1} + X_{9C2} + X_{9C3} + X_{9C4} + X_{9C5} + X_{9C6} + X_{9C7} + X_{9C8} + X_{9C9} + X_{9C10} + X_{9C11} + X_{9C12} + X_{9C13} + X_{9C14} + X_{9C15} = 1$
c 10 : $X_{10C1} + X_{10C2} + X_{10C3} + X_{10C4} + X_{10C5} + X_{10C6} + X_{10C7} + X_{10C8} + X_{10C9} + X_{10C10} + X_{10C11} + X_{10C12} + X_{10C13} + X_{10C14} + X_{10C15} = 1$
c 11 : $X_{11C1} + X_{11C2} + X_{11C3} + X_{11C4} + X_{11C5} + X_{11C6} + X_{11C7} + X_{11C8} + X_{11C9} + X_{11C10} + X_{11C11} + X_{11C12} + X_{11C13} + X_{11C14} + X_{11C15} = 1$

These are the Resolution III constraints. These constraints prevent two or more first-order terms (X_1 - X_8) from being assigned to the same column. The columns are of the

underlying full factorial design: c1 is “a”, c2 is “b”, c3 is “ab”, c4 is “c”, c5 is “ac”, etc. There are 15 constraints required; one for column one through column fifteen.

c 12 : $X1C1 + X2C1 + X4C1 + X8C1 + X14C1 + X13C1 + X7C1 + X11C1 \leq 1$
c 13 : $X1C2 + X2C2 + X4C2 + X8C2 + X14C2 + X13C2 + X7C2 + X11C2 \leq 1$
c 14 : $X1C3 + X2C3 + X4C3 + X8C3 + X14C3 + X13C3 + X7C3 + X11C3 \leq 1$
c 15 : $X1C4 + X2C4 + X4C4 + X8C4 + X14C4 + X13C4 + X7C4 + X11C4 \leq 1$
c 16 : $X1C5 + X2C5 + X4C5 + X8C5 + X14C5 + X13C5 + X7C5 + X11C5 \leq 1$
c 17 : $X1C6 + X2C6 + X4C6 + X8C6 + X14C6 + X13C6 + X7C6 + X11C6 \leq 1$
c 18 : $X1C7 + X2C7 + X4C7 + X8C7 + X14C7 + X13C7 + X7C7 + X11C7 \leq 1$
c 19 : $X1C8 + X2C8 + X4C8 + X8C8 + X14C8 + X13C8 + X7C8 + X11C8 \leq 1$
c 20 : $X1C9 + X2C9 + X4C9 + X8C9 + X14C9 + X13C9 + X7C9 + X11C9 \leq 1$
c 21 : $X1C10 + X2C10 + X4C10 + X8C10 + X14C10 + X13C10 + X7C10 + X11C10 \leq 1$
c 22 : $X1C11 + X2C11 + X4C11 + X8C11 + X14C11 + X13C11 + X7C11 + X11C11 \leq 1$
c 23 : $X1C12 + X2C12 + X4C12 + X8C12 + X14C12 + X13C12 + X7C12 + X11C12 \leq 1$
c 24 : $X1C13 + X2C13 + X4C13 + X8C13 + X14C13 + X13C13 + X7C13 + X11C13 \leq 1$
c 25 : $X1C14 + X2C14 + X4C14 + X8C14 + X14C14 + X13C14 + X7C14 + X11C14 \leq 1$
c 26 : $X1C15 + X2C15 + X4C15 + X8C15 + X14C15 + X13C15 + X7C15 + X11C15 \leq 1$

This next set of constraints is the confounding constraints. The IP will try to force each $X_i = 0$. Each X_i must not be set to zero if the i th term is confounded with any other term. These constraints check beginning with factor A and column one. If factor A is assigned to column 1 ($X1C1=1$) and if some other term (an interaction term) is also assigned to column 1, then $X1$ must equal 1, otherwise it is not constrained to be either zero or one. For this example there are 11 x 15 (or 165) of these constraints.

c 27 : $-15 X1 + 15 X1C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 \leq 16$
c 28 : $-15 X1 + 15 X1C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 \leq 16$
c 29 : $-15 X1 + 15 X1C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 \leq 16$
c 30 : $-15 X1 + 15 X1C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 \leq 16$
c 31 : $-15 X1 + 15 X1C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 \leq 16$
c 32 : $-15 X1 + 15 X1C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 \leq 16$
c 33 : $-15 X1 + 15 X1C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 \leq 16$
c 34 : $-15 X1 + 15 X1C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 \leq 16$
c 35 : $-15 X1 + 15 X1C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 \leq 16$
c 36 : $-15 X1 + 15 X1C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 \leq 16$

c 37 : - 15 X1 + 15 X1C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11
 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 38 : - 15 X1 + 15 X1C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12
 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 39 : - 15 X1 + 15 X1C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13
 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 40 : - 15 X1 + 15 X1C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14
 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 41 : - 15 X1 + 15 X1C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15
 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 42 : - 15 X2 + 15 X2C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 +
 X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 43 : - 15 X2 + 15 X2C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 +
 X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 44 : - 15 X2 + 15 X2C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 +
 X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 45 : - 15 X2 + 15 X2C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 +
 X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 46 : - 15 X2 + 15 X2C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 +
 X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 47 : - 15 X2 + 15 X2C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 +
 X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 48 : - 15 X2 + 15 X2C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 +
 X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 49 : - 15 X2 + 15 X2C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 +
 X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 50 : - 15 X2 + 15 X2C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 +
 X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 51 : - 15 X2 + 15 X2C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10
 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 52 : - 15 X2 + 15 X2C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11
 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 53 : - 15 X2 + 15 X2C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12
 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 54 : - 15 X2 + 15 X2C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13
 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 55 : - 15 X2 + 15 X2C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14
 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 56 : - 15 X2 + 15 X2C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15
 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 57 : - 15 X3 + 15 X3C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 +
 X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 58 : - 15 X3 + 15 X3C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 +
 X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 59 : - 15 X3 + 15 X3C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 +
 X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 60 : - 15 X3 + 15 X3C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 +
 X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 61 : - 15 X3 + 15 X3C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 +
 X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 62 : - 15 X3 + 15 X3C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 +
 X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 63 : - 15 X3 + 15 X3C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 +
 X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 64 : - 15 X3 + 15 X3C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 +
 X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16

c 65 : - 15 X3 + 15 X3C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 66 : - 15 X3 + 15 X3C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 67 : - 15 X3 + 15 X3C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 68 : - 15 X3 + 15 X3C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 69 : - 15 X3 + 15 X3C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 70 : - 15 X3 + 15 X3C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 71 : - 15 X3 + 15 X3C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 72 : - 15 X4 + 15 X4C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 73 : - 15 X4 + 15 X4C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 74 : - 15 X4 + 15 X4C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 75 : - 15 X4 + 15 X4C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 76 : - 15 X4 + 15 X4C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 77 : - 15 X4 + 15 X4C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 78 : - 15 X4 + 15 X4C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 79 : - 15 X4 + 15 X4C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 80 : - 15 X4 + 15 X4C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 81 : - 15 X4 + 15 X4C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 82 : - 15 X4 + 15 X4C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 83 : - 15 X4 + 15 X4C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 84 : - 15 X4 + 15 X4C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 85 : - 15 X4 + 15 X4C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 86 : - 15 X4 + 15 X4C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 87 : - 15 X5 + 15 X5C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 88 : - 15 X5 + 15 X5C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 89 : - 15 X5 + 15 X5C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 90 : - 15 X5 + 15 X5C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 91 : - 15 X5 + 15 X5C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 92 : - 15 X5 + 15 X5C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16

c 93 : - 15 X5 + 15 X5C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 94 : - 15 X5 + 15 X5C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 95 : - 15 X5 + 15 X5C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 96 : - 15 X5 + 15 X5C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 97 : - 15 X5 + 15 X5C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 98 : - 15 X5 + 15 X5C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 99 : - 15 X5 + 15 X5C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 100 : - 15 X5 + 15 X5C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 101 : - 15 X5 + 15 X5C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 102 : - 15 X6 + 15 X6C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 103 : - 15 X6 + 15 X6C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 104 : - 15 X6 + 15 X6C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 105 : - 15 X6 + 15 X6C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 106 : - 15 X6 + 15 X6C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 107 : - 15 X6 + 15 X6C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 108 : - 15 X6 + 15 X6C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 109 : - 15 X6 + 15 X6C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 110 : - 15 X6 + 15 X6C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 111 : - 15 X6 + 15 X6C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 112 : - 15 X6 + 15 X6C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 113 : - 15 X6 + 15 X6C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 114 : - 15 X6 + 15 X6C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 115 : - 15 X6 + 15 X6C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 116 : - 15 X6 + 15 X6C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 117 : - 15 X7 + 15 X7C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 118 : - 15 X7 + 15 X7C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 119 : - 15 X7 + 15 X7C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 120 : - 15 X7 + 15 X7C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16

c 121 : - 15 X7 + 15 X7C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 +
 X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 122 : - 15 X7 + 15 X7C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 +
 X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 123 : - 15 X7 + 15 X7C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 +
 X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 124 : - 15 X7 + 15 X7C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 +
 X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 125 : - 15 X7 + 15 X7C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 +
 X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 126 : - 15 X7 + 15 X7C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 +
 X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 127 : - 15 X7 + 15 X7C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 +
 X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 128 : - 15 X7 + 15 X7C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 +
 X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 129 : - 15 X7 + 15 X7C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 +
 X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 130 : - 15 X7 + 15 X7C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 +
 X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 131 : - 15 X7 + 15 X7C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 +
 X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 132 : - 15 X8 + 15 X8C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 +
 X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 133 : - 15 X8 + 15 X8C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 +
 X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 134 : - 15 X8 + 15 X8C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 +
 X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 135 : - 15 X8 + 15 X8C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 +
 X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 136 : - 15 X8 + 15 X8C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 +
 X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 137 : - 15 X8 + 15 X8C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 +
 X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 138 : - 15 X8 + 15 X8C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 +
 X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 139 : - 15 X8 + 15 X8C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 +
 X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 140 : - 15 X8 + 15 X8C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 +
 X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 141 : - 15 X8 + 15 X8C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 +
 X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 142 : - 15 X8 + 15 X8C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 +
 X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 143 : - 15 X8 + 15 X8C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 +
 X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 144 : - 15 X8 + 15 X8C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 +
 X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 145 : - 15 X8 + 15 X8C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 +
 X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 146 : - 15 X8 + 15 X8C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 +
 X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 147 : - 15 X9 + 15 X9C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 +
 X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 148 : - 15 X9 + 15 X9C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 +
 X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16

c 149 : - 15 X9 + 15 X9C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 150 : - 15 X9 + 15 X9C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 151 : - 15 X9 + 15 X9C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 152 : - 15 X9 + 15 X9C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 153 : - 15 X9 + 15 X9C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 154 : - 15 X9 + 15 X9C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 155 : - 15 X9 + 15 X9C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 156 : - 15 X9 + 15 X9C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 157 : - 15 X9 + 15 X9C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 158 : - 15 X9 + 15 X9C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 159 : - 15 X9 + 15 X9C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 160 : - 15 X9 + 15 X9C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 161 : - 15 X9 + 15 X9C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16
 c 162 : - 15 X10 + 15 X10C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 163 : - 15 X10 + 15 X10C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 164 : - 15 X10 + 15 X10C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 165 : - 15 X10 + 15 X10C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 166 : - 15 X10 + 15 X10C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 167 : - 15 X10 + 15 X10C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 168 : - 15 X10 + 15 X10C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 169 : - 15 X10 + 15 X10C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 170 : - 15 X10 + 15 X10C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 171 : - 15 X10 + 15 X10C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 172 : - 15 X10 + 15 X10C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 173 : - 15 X10 + 15 X10C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 174 : - 15 X10 + 15 X10C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 175 : - 15 X10 + 15 X10C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 176 : - 15 X10 + 15 X10C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16

c 177 : - 15 X11 + 15 X11C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 <= 16
 c 178 : - 15 X11 + 15 X11C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 <= 16
 c 179 : - 15 X11 + 15 X11C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 <= 16
 c 180 : - 15 X11 + 15 X11C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 <= 16
 c 181 : - 15 X11 + 15 X11C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 <= 16
 c 182 : - 15 X11 + 15 X11C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 <= 16
 c 183 : - 15 X11 + 15 X11C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 <= 16
 c 184 : - 15 X11 + 15 X11C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 <= 16
 c 185 : - 15 X11 + 15 X11C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 <= 16
 c 186 : - 15 X11 + 15 X11C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 <= 16
 c 187 : - 15 X11 + 15 X11C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 <= 16
 c 188 : - 15 X11 + 15 X11C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 + X14C12 + X15C12 <= 16
 c 189 : - 15 X11 + 15 X11C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 <= 16
 c 190 : - 15 X11 + 15 X11C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 <= 16
 c 191 : - 15 X11 + 15 X11C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 <= 16

The interaction terms are not free to be assigned to any column. If A is assigned to column 1 and B is assigned to column 2, then the AB term MUST be assigned to column 3. Consider constraint 192. If A is assigned to column 1 and B is assigned to column 2, then AB (X9) must be assigned to column 3 (X9C3 = 1). If X1 or X2 are assigned to any other column, then the constraint is satisfied and X9C3 is unconstrained. In this example, there are 15x14 choices (210) for each two-factor interaction term. Three interaction terms mean that 630 constraints of this type are needed for this problem.

c 192 : X1C1 + X2C2 - 2 X9C3 <= 1
 c 193 : X1C1 + X2C3 - 2 X9C2 <= 1
 c 194 : X1C1 + X2C4 - 2 X9C5 <= 1
 c 195 : X1C1 + X2C5 - 2 X9C4 <= 1
 c 196 : X1C1 + X2C6 - 2 X9C7 <= 1
 c 197 : X1C1 + X2C7 - 2 X9C6 <= 1
 c 198 : X1C1 + X2C8 - 2 X9C9 <= 1

c 199 : $X1C1 + X2C9 - 2 X9C8 \leq 1$
 c 200 : $X1C1 + X2C10 - 2 X9C11 \leq 1$
 c 201 : $X1C1 + X2C11 - 2 X9C10 \leq 1$
 c 202 : $X1C1 + X2C12 - 2 X9C13 \leq 1$
 c 203 : $X1C1 + X2C13 - 2 X9C12 \leq 1$
 c 204 : $X1C1 + X2C14 - 2 X9C15 \leq 1$
 c 205 : $X1C1 + X2C15 - 2 X9C14 \leq 1$
 c 206 : $X1C2 + X2C1 - 2 X9C3 \leq 1$
 c 207 : $X1C2 + X2C3 - 2 X9C1 \leq 1$
 c 208 : $X1C2 + X2C4 - 2 X9C6 \leq 1$
 c 209 : $X1C2 + X2C5 - 2 X9C7 \leq 1$
 c 210 : $X1C2 + X2C6 - 2 X9C4 \leq 1$
 c 211 : $X1C2 + X2C7 - 2 X9C5 \leq 1$
 c 212 : $X1C2 + X2C8 - 2 X9C10 \leq 1$
 c 213 : $X1C2 + X2C9 - 2 X9C11 \leq 1$
 c 214 : $X1C2 + X2C10 - 2 X9C8 \leq 1$
 c 215 : $X1C2 + X2C11 - 2 X9C9 \leq 1$
 c 216 : $X1C2 + X2C12 - 2 X9C14 \leq 1$
 c 217 : $X1C2 + X2C13 - 2 X9C15 \leq 1$
 c 218 : $X1C2 + X2C14 - 2 X9C12 \leq 1$
 c 219 : $X1C2 + X2C15 - 2 X9C13 \leq 1$
 c 220 : $X1C3 + X2C1 - 2 X9C2 \leq 1$
 c 221 : $X1C3 + X2C2 - 2 X9C1 \leq 1$
 c 222 : $X1C3 + X2C4 - 2 X9C7 \leq 1$
 c 223 : $X1C3 + X2C5 - 2 X9C6 \leq 1$
 c 224 : $X1C3 + X2C6 - 2 X9C5 \leq 1$
 c 225 : $X1C3 + X2C7 - 2 X9C4 \leq 1$
 c 226 : $X1C3 + X2C8 - 2 X9C11 \leq 1$
 c 227 : $X1C3 + X2C9 - 2 X9C10 \leq 1$
 c 228 : $X1C3 + X2C10 - 2 X9C9 \leq 1$
 c 229 : $X1C3 + X2C11 - 2 X9C8 \leq 1$
 c 230 : $X1C3 + X2C12 - 2 X9C15 \leq 1$
 c 231 : $X1C3 + X2C13 - 2 X9C14 \leq 1$
 c 232 : $X1C3 + X2C14 - 2 X9C13 \leq 1$
 c 233 : $X1C3 + X2C15 - 2 X9C12 \leq 1$
 c 234 : $X1C4 + X2C1 - 2 X9C5 \leq 1$
 c 235 : $X1C4 + X2C2 - 2 X9C6 \leq 1$
 c 236 : $X1C4 + X2C3 - 2 X9C7 \leq 1$
 c 237 : $X1C4 + X2C5 - 2 X9C1 \leq 1$
 c 238 : $X1C4 + X2C6 - 2 X9C2 \leq 1$
 c 239 : $X1C4 + X2C7 - 2 X9C3 \leq 1$
 c 240 : $X1C4 + X2C8 - 2 X9C12 \leq 1$
 c 241 : $X1C4 + X2C9 - 2 X9C13 \leq 1$
 c 242 : $X1C4 + X2C10 - 2 X9C14 \leq 1$
 c 243 : $X1C4 + X2C11 - 2 X9C15 \leq 1$
 c 244 : $X1C4 + X2C12 - 2 X9C8 \leq 1$
 c 245 : $X1C4 + X2C13 - 2 X9C9 \leq 1$
 c 246 : $X1C4 + X2C14 - 2 X9C10 \leq 1$
 c 247 : $X1C4 + X2C15 - 2 X9C11 \leq 1$
 c 248 : $X1C5 + X2C1 - 2 X9C4 \leq 1$
 c 249 : $X1C5 + X2C2 - 2 X9C7 \leq 1$
 c 250 : $X1C5 + X2C3 - 2 X9C6 \leq 1$
 c 251 : $X1C5 + X2C4 - 2 X9C1 \leq 1$
 c 252 : $X1C5 + X2C6 - 2 X9C3 \leq 1$
 c 253 : $X1C5 + X2C7 - 2 X9C2 \leq 1$
 c 254 : $X1C5 + X2C8 - 2 X9C13 \leq 1$

c 255 : $X1C5 + X2C9 - 2 X9C12 \leq 1$
c 256 : $X1C5 + X2C10 - 2 X9C15 \leq 1$
c 257 : $X1C5 + X2C11 - 2 X9C14 \leq 1$
c 258 : $X1C5 + X2C12 - 2 X9C9 \leq 1$
c 259 : $X1C5 + X2C13 - 2 X9C8 \leq 1$
c 260 : $X1C5 + X2C14 - 2 X9C11 \leq 1$
c 261 : $X1C5 + X2C15 - 2 X9C10 \leq 1$
c 262 : $X1C6 + X2C1 - 2 X9C7 \leq 1$
c 263 : $X1C6 + X2C2 - 2 X9C4 \leq 1$
c 264 : $X1C6 + X2C3 - 2 X9C5 \leq 1$
c 265 : $X1C6 + X2C4 - 2 X9C2 \leq 1$
c 266 : $X1C6 + X2C5 - 2 X9C3 \leq 1$
c 267 : $X1C6 + X2C7 - 2 X9C1 \leq 1$
c 268 : $X1C6 + X2C8 - 2 X9C14 \leq 1$
c 269 : $X1C6 + X2C9 - 2 X9C15 \leq 1$
c 270 : $X1C6 + X2C10 - 2 X9C12 \leq 1$
c 271 : $X1C6 + X2C11 - 2 X9C13 \leq 1$
c 272 : $X1C6 + X2C12 - 2 X9C10 \leq 1$
c 273 : $X1C6 + X2C13 - 2 X9C11 \leq 1$
c 274 : $X1C6 + X2C14 - 2 X9C8 \leq 1$
c 275 : $X1C6 + X2C15 - 2 X9C9 \leq 1$
c 276 : $X1C7 + X2C1 - 2 X9C6 \leq 1$
c 277 : $X1C7 + X2C2 - 2 X9C5 \leq 1$
c 278 : $X1C7 + X2C3 - 2 X9C4 \leq 1$
c 279 : $X1C7 + X2C4 - 2 X9C3 \leq 1$
c 280 : $X1C7 + X2C5 - 2 X9C2 \leq 1$
c 281 : $X1C7 + X2C6 - 2 X9C1 \leq 1$
c 282 : $X1C7 + X2C8 - 2 X9C15 \leq 1$
c 283 : $X1C7 + X2C9 - 2 X9C14 \leq 1$
c 284 : $X1C7 + X2C10 - 2 X9C13 \leq 1$
c 285 : $X1C7 + X2C11 - 2 X9C12 \leq 1$
c 286 : $X1C7 + X2C12 - 2 X9C11 \leq 1$
c 287 : $X1C7 + X2C13 - 2 X9C10 \leq 1$
c 288 : $X1C7 + X2C14 - 2 X9C9 \leq 1$
c 289 : $X1C7 + X2C15 - 2 X9C8 \leq 1$
c 290 : $X1C8 + X2C1 - 2 X9C9 \leq 1$
c 291 : $X1C8 + X2C2 - 2 X9C10 \leq 1$
c 292 : $X1C8 + X2C3 - 2 X9C11 \leq 1$
c 293 : $X1C8 + X2C4 - 2 X9C12 \leq 1$
c 294 : $X1C8 + X2C5 - 2 X9C13 \leq 1$
c 295 : $X1C8 + X2C6 - 2 X9C14 \leq 1$
c 296 : $X1C8 + X2C7 - 2 X9C15 \leq 1$
c 297 : $X1C8 + X2C9 - 2 X9C1 \leq 1$
c 298 : $X1C8 + X2C10 - 2 X9C2 \leq 1$
c 299 : $X1C8 + X2C11 - 2 X9C3 \leq 1$
c 300 : $X1C8 + X2C12 - 2 X9C4 \leq 1$
c 301 : $X1C8 + X2C13 - 2 X9C5 \leq 1$
c 302 : $X1C8 + X2C14 - 2 X9C6 \leq 1$
c 303 : $X1C8 + X2C15 - 2 X9C7 \leq 1$
c 304 : $X1C9 + X2C1 - 2 X9C8 \leq 1$
c 305 : $X1C9 + X2C2 - 2 X9C11 \leq 1$
c 306 : $X1C9 + X2C3 - 2 X9C10 \leq 1$
c 307 : $X1C9 + X2C4 - 2 X9C13 \leq 1$
c 308 : $X1C9 + X2C5 - 2 X9C12 \leq 1$
c 309 : $X1C9 + X2C6 - 2 X9C15 \leq 1$
c 310 : $X1C9 + X2C7 - 2 X9C14 \leq 1$

c 311: $X1C9 + X2C8 - 2 X9C1 \leq 1$
 c 312: $X1C9 + X2C10 - 2 X9C3 \leq 1$
 c 313: $X1C9 + X2C11 - 2 X9C2 \leq 1$
 c 314: $X1C9 + X2C12 - 2 X9C5 \leq 1$
 c 315: $X1C9 + X2C13 - 2 X9C4 \leq 1$
 c 316: $X1C9 + X2C14 - 2 X9C7 \leq 1$
 c 317: $X1C9 + X2C15 - 2 X9C6 \leq 1$
 c 318: $X1C10 + X2C1 - 2 X9C11 \leq 1$
 c 319: $X1C10 + X2C2 - 2 X9C8 \leq 1$
 c 320: $X1C10 + X2C3 - 2 X9C9 \leq 1$
 c 321: $X1C10 + X2C4 - 2 X9C14 \leq 1$
 c 322: $X1C10 + X2C5 - 2 X9C15 \leq 1$
 c 323: $X1C10 + X2C6 - 2 X9C12 \leq 1$
 c 324: $X1C10 + X2C7 - 2 X9C13 \leq 1$
 c 325: $X1C10 + X2C8 - 2 X9C2 \leq 1$
 c 326: $X1C10 + X2C9 - 2 X9C3 \leq 1$
 c 327: $X1C10 + X2C11 - 2 X9C1 \leq 1$
 c 328: $X1C10 + X2C12 - 2 X9C6 \leq 1$
 c 329: $X1C10 + X2C13 - 2 X9C7 \leq 1$
 c 330: $X1C10 + X2C14 - 2 X9C4 \leq 1$
 c 331: $X1C10 + X2C15 - 2 X9C5 \leq 1$
 c 332: $X1C11 + X2C1 - 2 X9C10 \leq 1$
 c 333: $X1C11 + X2C2 - 2 X9C9 \leq 1$
 c 334: $X1C11 + X2C3 - 2 X9C8 \leq 1$
 c 335: $X1C11 + X2C4 - 2 X9C15 \leq 1$
 c 336: $X1C11 + X2C5 - 2 X9C14 \leq 1$
 c 337: $X1C11 + X2C6 - 2 X9C13 \leq 1$
 c 338: $X1C11 + X2C7 - 2 X9C12 \leq 1$
 c 339: $X1C11 + X2C8 - 2 X9C3 \leq 1$
 c 340: $X1C11 + X2C9 - 2 X9C2 \leq 1$
 c 341: $X1C11 + X2C10 - 2 X9C1 \leq 1$
 c 342: $X1C11 + X2C12 - 2 X9C7 \leq 1$
 c 343: $X1C11 + X2C13 - 2 X9C6 \leq 1$
 c 344: $X1C11 + X2C14 - 2 X9C5 \leq 1$
 c 345: $X1C11 + X2C15 - 2 X9C4 \leq 1$
 c 346: $X1C12 + X2C1 - 2 X9C13 \leq 1$
 c 347: $X1C12 + X2C2 - 2 X9C14 \leq 1$
 c 348: $X1C12 + X2C3 - 2 X9C15 \leq 1$
 c 349: $X1C12 + X2C4 - 2 X9C8 \leq 1$
 c 350: $X1C12 + X2C5 - 2 X9C9 \leq 1$
 c 351: $X1C12 + X2C6 - 2 X9C10 \leq 1$
 c 352: $X1C12 + X2C7 - 2 X9C11 \leq 1$
 c 353: $X1C12 + X2C8 - 2 X9C4 \leq 1$
 c 354: $X1C12 + X2C9 - 2 X9C5 \leq 1$
 c 355: $X1C12 + X2C10 - 2 X9C6 \leq 1$
 c 356: $X1C12 + X2C11 - 2 X9C7 \leq 1$
 c 357: $X1C12 + X2C13 - 2 X9C1 \leq 1$
 c 358: $X1C12 + X2C14 - 2 X9C2 \leq 1$
 c 359: $X1C12 + X2C15 - 2 X9C3 \leq 1$
 c 360: $X1C13 + X2C1 - 2 X9C12 \leq 1$
 c 361: $X1C13 + X2C2 - 2 X9C15 \leq 1$
 c 362: $X1C13 + X2C3 - 2 X9C14 \leq 1$
 c 363: $X1C13 + X2C4 - 2 X9C9 \leq 1$
 c 364: $X1C13 + X2C5 - 2 X9C8 \leq 1$
 c 365: $X1C13 + X2C6 - 2 X9C11 \leq 1$
 c 366: $X1C13 + X2C7 - 2 X9C10 \leq 1$

c 367 : $X1C13 + X2C8 - 2 X9C5 \leq 1$
c 368 : $X1C13 + X2C9 - 2 X9C4 \leq 1$
c 369 : $X1C13 + X2C10 - 2 X9C7 \leq 1$
c 370 : $X1C13 + X2C11 - 2 X9C6 \leq 1$
c 371 : $X1C13 + X2C12 - 2 X9C1 \leq 1$
c 372 : $X1C13 + X2C14 - 2 X9C3 \leq 1$
c 373 : $X1C13 + X2C15 - 2 X9C2 \leq 1$
c 374 : $X1C14 + X2C1 - 2 X9C15 \leq 1$
c 375 : $X1C14 + X2C2 - 2 X9C12 \leq 1$
c 376 : $X1C14 + X2C3 - 2 X9C13 \leq 1$
c 377 : $X1C14 + X2C4 - 2 X9C10 \leq 1$
c 378 : $X1C14 + X2C5 - 2 X9C11 \leq 1$
c 379 : $X1C14 + X2C6 - 2 X9C8 \leq 1$
c 380 : $X1C14 + X2C7 - 2 X9C9 \leq 1$
c 381 : $X1C14 + X2C8 - 2 X9C6 \leq 1$
c 382 : $X1C14 + X2C9 - 2 X9C7 \leq 1$
c 383 : $X1C14 + X2C10 - 2 X9C4 \leq 1$
c 384 : $X1C14 + X2C11 - 2 X9C5 \leq 1$
c 385 : $X1C14 + X2C12 - 2 X9C2 \leq 1$
c 386 : $X1C14 + X2C13 - 2 X9C3 \leq 1$
c 387 : $X1C14 + X2C15 - 2 X9C1 \leq 1$
c 388 : $X1C15 + X2C1 - 2 X9C14 \leq 1$
c 389 : $X1C15 + X2C2 - 2 X9C13 \leq 1$
c 390 : $X1C15 + X2C3 - 2 X9C12 \leq 1$
c 391 : $X1C15 + X2C4 - 2 X9C11 \leq 1$
c 392 : $X1C15 + X2C5 - 2 X9C10 \leq 1$
c 393 : $X1C15 + X2C6 - 2 X9C9 \leq 1$
c 394 : $X1C15 + X2C7 - 2 X9C8 \leq 1$
c 395 : $X1C15 + X2C8 - 2 X9C7 \leq 1$
c 396 : $X1C15 + X2C9 - 2 X9C6 \leq 1$
c 397 : $X1C15 + X2C10 - 2 X9C5 \leq 1$
c 398 : $X1C15 + X2C11 - 2 X9C4 \leq 1$
c 399 : $X1C15 + X2C12 - 2 X9C3 \leq 1$
c 400 : $X1C15 + X2C13 - 2 X9C2 \leq 1$
c 401 : $X1C15 + X2C14 - 2 X9C1 \leq 1$
c 402 : $X1C1 + X4C2 - 2 X10C3 \leq 1$
c 403 : $X1C1 + X4C3 - 2 X10C2 \leq 1$
c 404 : $X1C1 + X4C4 - 2 X10C5 \leq 1$
c 405 : $X1C1 + X4C5 - 2 X10C4 \leq 1$
c 406 : $X1C1 + X4C6 - 2 X10C7 \leq 1$
c 407 : $X1C1 + X4C7 - 2 X10C6 \leq 1$
c 408 : $X1C1 + X4C8 - 2 X10C9 \leq 1$
c 409 : $X1C1 + X4C9 - 2 X10C8 \leq 1$
c 410 : $X1C1 + X4C10 - 2 X10C11 \leq 1$
c 411 : $X1C1 + X4C11 - 2 X10C10 \leq 1$
c 412 : $X1C1 + X4C12 - 2 X10C13 \leq 1$
c 413 : $X1C1 + X4C13 - 2 X10C12 \leq 1$
c 414 : $X1C1 + X4C14 - 2 X10C15 \leq 1$
c 415 : $X1C1 + X4C15 - 2 X10C14 \leq 1$
c 416 : $X1C2 + X4C1 - 2 X10C3 \leq 1$
c 417 : $X1C2 + X4C3 - 2 X10C1 \leq 1$
c 418 : $X1C2 + X4C4 - 2 X10C6 \leq 1$
c 419 : $X1C2 + X4C5 - 2 X10C7 \leq 1$
c 420 : $X1C2 + X4C6 - 2 X10C4 \leq 1$
c 421 : $X1C2 + X4C7 - 2 X10C5 \leq 1$
c 422 : $X1C2 + X4C8 - 2 X10C10 \leq 1$

c 423 : $X1C2 + X4C9 - 2 X10C11 \leq 1$
c 424 : $X1C2 + X4C10 - 2 X10C8 \leq 1$
c 425 : $X1C2 + X4C11 - 2 X10C9 \leq 1$
c 426 : $X1C2 + X4C12 - 2 X10C14 \leq 1$
c 427 : $X1C2 + X4C13 - 2 X10C15 \leq 1$
c 428 : $X1C2 + X4C14 - 2 X10C12 \leq 1$
c 429 : $X1C2 + X4C15 - 2 X10C13 \leq 1$
c 430 : $X1C3 + X4C1 - 2 X10C2 \leq 1$
c 431 : $X1C3 + X4C2 - 2 X10C1 \leq 1$
c 432 : $X1C3 + X4C4 - 2 X10C7 \leq 1$
c 433 : $X1C3 + X4C5 - 2 X10C6 \leq 1$
c 434 : $X1C3 + X4C6 - 2 X10C5 \leq 1$
c 435 : $X1C3 + X4C7 - 2 X10C4 \leq 1$
c 436 : $X1C3 + X4C8 - 2 X10C11 \leq 1$
c 437 : $X1C3 + X4C9 - 2 X10C10 \leq 1$
c 438 : $X1C3 + X4C10 - 2 X10C9 \leq 1$
c 439 : $X1C3 + X4C11 - 2 X10C8 \leq 1$
c 440 : $X1C3 + X4C12 - 2 X10C15 \leq 1$
c 441 : $X1C3 + X4C13 - 2 X10C14 \leq 1$
c 442 : $X1C3 + X4C14 - 2 X10C13 \leq 1$
c 443 : $X1C3 + X4C15 - 2 X10C12 \leq 1$
c 444 : $X1C4 + X4C1 - 2 X10C5 \leq 1$
c 445 : $X1C4 + X4C2 - 2 X10C6 \leq 1$
c 446 : $X1C4 + X4C3 - 2 X10C7 \leq 1$
c 447 : $X1C4 + X4C5 - 2 X10C1 \leq 1$
c 448 : $X1C4 + X4C6 - 2 X10C2 \leq 1$
c 449 : $X1C4 + X4C7 - 2 X10C3 \leq 1$
c 450 : $X1C4 + X4C8 - 2 X10C12 \leq 1$
c 451 : $X1C4 + X4C9 - 2 X10C13 \leq 1$
c 452 : $X1C4 + X4C10 - 2 X10C14 \leq 1$
c 453 : $X1C4 + X4C11 - 2 X10C15 \leq 1$
c 454 : $X1C4 + X4C12 - 2 X10C8 \leq 1$
c 455 : $X1C4 + X4C13 - 2 X10C9 \leq 1$
c 456 : $X1C4 + X4C14 - 2 X10C10 \leq 1$
c 457 : $X1C4 + X4C15 - 2 X10C11 \leq 1$
c 458 : $X1C5 + X4C1 - 2 X10C4 \leq 1$
c 459 : $X1C5 + X4C2 - 2 X10C7 \leq 1$
c 460 : $X1C5 + X4C3 - 2 X10C6 \leq 1$
c 461 : $X1C5 + X4C4 - 2 X10C1 \leq 1$
c 462 : $X1C5 + X4C6 - 2 X10C3 \leq 1$
c 463 : $X1C5 + X4C7 - 2 X10C2 \leq 1$
c 464 : $X1C5 + X4C8 - 2 X10C13 \leq 1$
c 465 : $X1C5 + X4C9 - 2 X10C12 \leq 1$
c 466 : $X1C5 + X4C10 - 2 X10C15 \leq 1$
c 467 : $X1C5 + X4C11 - 2 X10C14 \leq 1$
c 468 : $X1C5 + X4C12 - 2 X10C9 \leq 1$
c 469 : $X1C5 + X4C13 - 2 X10C8 \leq 1$
c 470 : $X1C5 + X4C14 - 2 X10C11 \leq 1$
c 471 : $X1C5 + X4C15 - 2 X10C10 \leq 1$
c 472 : $X1C6 + X4C1 - 2 X10C7 \leq 1$
c 473 : $X1C6 + X4C2 - 2 X10C4 \leq 1$
c 474 : $X1C6 + X4C3 - 2 X10C5 \leq 1$
c 475 : $X1C6 + X4C4 - 2 X10C2 \leq 1$
c 476 : $X1C6 + X4C5 - 2 X10C3 \leq 1$
c 477 : $X1C6 + X4C7 - 2 X10C1 \leq 1$
c 478 : $X1C6 + X4C8 - 2 X10C14 \leq 1$

c 479: $X1C6 + X4C9 - 2 X10C15 \leq 1$
c 480: $X1C6 + X4C10 - 2 X10C12 \leq 1$
c 481: $X1C6 + X4C11 - 2 X10C13 \leq 1$
c 482: $X1C6 + X4C12 - 2 X10C10 \leq 1$
c 483: $X1C6 + X4C13 - 2 X10C11 \leq 1$
c 484: $X1C6 + X4C14 - 2 X10C8 \leq 1$
c 485: $X1C6 + X4C15 - 2 X10C9 \leq 1$
c 486: $X1C7 + X4C1 - 2 X10C6 \leq 1$
c 487: $X1C7 + X4C2 - 2 X10C5 \leq 1$
c 488: $X1C7 + X4C3 - 2 X10C4 \leq 1$
c 489: $X1C7 + X4C4 - 2 X10C3 \leq 1$
c 490: $X1C7 + X4C5 - 2 X10C2 \leq 1$
c 491: $X1C7 + X4C6 - 2 X10C1 \leq 1$
c 492: $X1C7 + X4C8 - 2 X10C15 \leq 1$
c 493: $X1C7 + X4C9 - 2 X10C14 \leq 1$
c 494: $X1C7 + X4C10 - 2 X10C13 \leq 1$
c 495: $X1C7 + X4C11 - 2 X10C12 \leq 1$
c 496: $X1C7 + X4C12 - 2 X10C11 \leq 1$
c 497: $X1C7 + X4C13 - 2 X10C10 \leq 1$
c 498: $X1C7 + X4C14 - 2 X10C9 \leq 1$
c 499: $X1C7 + X4C15 - 2 X10C8 \leq 1$
c 500: $X1C8 + X4C1 - 2 X10C9 \leq 1$
c 501: $X1C8 + X4C2 - 2 X10C10 \leq 1$
c 502: $X1C8 + X4C3 - 2 X10C11 \leq 1$
c 503: $X1C8 + X4C4 - 2 X10C12 \leq 1$
c 504: $X1C8 + X4C5 - 2 X10C13 \leq 1$
c 505: $X1C8 + X4C6 - 2 X10C14 \leq 1$
c 506: $X1C8 + X4C7 - 2 X10C15 \leq 1$
c 507: $X1C8 + X4C9 - 2 X10C1 \leq 1$
c 508: $X1C8 + X4C10 - 2 X10C2 \leq 1$
c 509: $X1C8 + X4C11 - 2 X10C3 \leq 1$
c 510: $X1C8 + X4C12 - 2 X10C4 \leq 1$
c 511: $X1C8 + X4C13 - 2 X10C5 \leq 1$
c 512: $X1C8 + X4C14 - 2 X10C6 \leq 1$
c 513: $X1C8 + X4C15 - 2 X10C7 \leq 1$
c 514: $X1C9 + X4C1 - 2 X10C8 \leq 1$
c 515: $X1C9 + X4C2 - 2 X10C11 \leq 1$
c 516: $X1C9 + X4C3 - 2 X10C10 \leq 1$
c 517: $X1C9 + X4C4 - 2 X10C13 \leq 1$
c 518: $X1C9 + X4C5 - 2 X10C12 \leq 1$
c 519: $X1C9 + X4C6 - 2 X10C15 \leq 1$
c 520: $X1C9 + X4C7 - 2 X10C14 \leq 1$
c 521: $X1C9 + X4C8 - 2 X10C1 \leq 1$
c 522: $X1C9 + X4C10 - 2 X10C3 \leq 1$
c 523: $X1C9 + X4C11 - 2 X10C2 \leq 1$
c 524: $X1C9 + X4C12 - 2 X10C5 \leq 1$
c 525: $X1C9 + X4C13 - 2 X10C4 \leq 1$
c 526: $X1C9 + X4C14 - 2 X10C7 \leq 1$
c 527: $X1C9 + X4C15 - 2 X10C6 \leq 1$
c 528: $X1C10 + X4C1 - 2 X10C11 \leq 1$
c 529: $X1C10 + X4C2 - 2 X10C8 \leq 1$
c 530: $X1C10 + X4C3 - 2 X10C9 \leq 1$
c 531: $X1C10 + X4C4 - 2 X10C14 \leq 1$
c 532: $X1C10 + X4C5 - 2 X10C15 \leq 1$
c 533: $X1C10 + X4C6 - 2 X10C12 \leq 1$
c 534: $X1C10 + X4C7 - 2 X10C13 \leq 1$

c 535 : $X1C10 + X4C8 - 2 X10C2 \leq 1$
c 536 : $X1C10 + X4C9 - 2 X10C3 \leq 1$
c 537 : $X1C10 + X4C11 - 2 X10C1 \leq 1$
c 538 : $X1C10 + X4C12 - 2 X10C6 \leq 1$
c 539 : $X1C10 + X4C13 - 2 X10C7 \leq 1$
c 540 : $X1C10 + X4C14 - 2 X10C4 \leq 1$
c 541 : $X1C10 + X4C15 - 2 X10C5 \leq 1$
c 542 : $X1C11 + X4C1 - 2 X10C10 \leq 1$
c 543 : $X1C11 + X4C2 - 2 X10C9 \leq 1$
c 544 : $X1C11 + X4C3 - 2 X10C8 \leq 1$
c 545 : $X1C11 + X4C4 - 2 X10C15 \leq 1$
c 546 : $X1C11 + X4C5 - 2 X10C14 \leq 1$
c 547 : $X1C11 + X4C6 - 2 X10C13 \leq 1$
c 548 : $X1C11 + X4C7 - 2 X10C12 \leq 1$
c 549 : $X1C11 + X4C8 - 2 X10C3 \leq 1$
c 550 : $X1C11 + X4C9 - 2 X10C2 \leq 1$
c 551 : $X1C11 + X4C10 - 2 X10C1 \leq 1$
c 552 : $X1C11 + X4C12 - 2 X10C7 \leq 1$
c 553 : $X1C11 + X4C13 - 2 X10C6 \leq 1$
c 554 : $X1C11 + X4C14 - 2 X10C5 \leq 1$
c 555 : $X1C11 + X4C15 - 2 X10C4 \leq 1$
c 556 : $X1C12 + X4C1 - 2 X10C13 \leq 1$
c 557 : $X1C12 + X4C2 - 2 X10C14 \leq 1$
c 558 : $X1C12 + X4C3 - 2 X10C15 \leq 1$
c 559 : $X1C12 + X4C4 - 2 X10C8 \leq 1$
c 560 : $X1C12 + X4C5 - 2 X10C9 \leq 1$
c 561 : $X1C12 + X4C6 - 2 X10C10 \leq 1$
c 562 : $X1C12 + X4C7 - 2 X10C11 \leq 1$
c 563 : $X1C12 + X4C8 - 2 X10C4 \leq 1$
c 564 : $X1C12 + X4C9 - 2 X10C5 \leq 1$
c 565 : $X1C12 + X4C10 - 2 X10C6 \leq 1$
c 566 : $X1C12 + X4C11 - 2 X10C7 \leq 1$
c 567 : $X1C12 + X4C13 - 2 X10C1 \leq 1$
c 568 : $X1C12 + X4C14 - 2 X10C2 \leq 1$
c 569 : $X1C12 + X4C15 - 2 X10C3 \leq 1$
c 570 : $X1C13 + X4C1 - 2 X10C12 \leq 1$
c 571 : $X1C13 + X4C2 - 2 X10C15 \leq 1$
c 572 : $X1C13 + X4C3 - 2 X10C14 \leq 1$
c 573 : $X1C13 + X4C4 - 2 X10C9 \leq 1$
c 574 : $X1C13 + X4C5 - 2 X10C8 \leq 1$
c 575 : $X1C13 + X4C6 - 2 X10C11 \leq 1$
c 576 : $X1C13 + X4C7 - 2 X10C10 \leq 1$
c 577 : $X1C13 + X4C8 - 2 X10C5 \leq 1$
c 578 : $X1C13 + X4C9 - 2 X10C4 \leq 1$
c 579 : $X1C13 + X4C10 - 2 X10C7 \leq 1$
c 580 : $X1C13 + X4C11 - 2 X10C6 \leq 1$
c 581 : $X1C13 + X4C12 - 2 X10C1 \leq 1$
c 582 : $X1C13 + X4C14 - 2 X10C3 \leq 1$
c 583 : $X1C13 + X4C15 - 2 X10C2 \leq 1$
c 584 : $X1C14 + X4C1 - 2 X10C15 \leq 1$
c 585 : $X1C14 + X4C2 - 2 X10C12 \leq 1$
c 586 : $X1C14 + X4C3 - 2 X10C13 \leq 1$
c 587 : $X1C14 + X4C4 - 2 X10C10 \leq 1$
c 588 : $X1C14 + X4C5 - 2 X10C11 \leq 1$
c 589 : $X1C14 + X4C6 - 2 X10C8 \leq 1$
c 590 : $X1C14 + X4C7 - 2 X10C9 \leq 1$

c 591 : $X1C14 + X4C8 - 2 X10C6 \leq 1$
c 592 : $X1C14 + X4C9 - 2 X10C7 \leq 1$
c 593 : $X1C14 + X4C10 - 2 X10C4 \leq 1$
c 594 : $X1C14 + X4C11 - 2 X10C5 \leq 1$
c 595 : $X1C14 + X4C12 - 2 X10C2 \leq 1$
c 596 : $X1C14 + X4C13 - 2 X10C3 \leq 1$
c 597 : $X1C14 + X4C15 - 2 X10C1 \leq 1$
c 598 : $X1C15 + X4C1 - 2 X10C14 \leq 1$
c 599 : $X1C15 + X4C2 - 2 X10C13 \leq 1$
c 600 : $X1C15 + X4C3 - 2 X10C12 \leq 1$
c 601 : $X1C15 + X4C4 - 2 X10C11 \leq 1$
c 602 : $X1C15 + X4C5 - 2 X10C10 \leq 1$
c 603 : $X1C15 + X4C6 - 2 X10C9 \leq 1$
c 604 : $X1C15 + X4C7 - 2 X10C8 \leq 1$
c 605 : $X1C15 + X4C8 - 2 X10C7 \leq 1$
c 606 : $X1C15 + X4C9 - 2 X10C6 \leq 1$
c 607 : $X1C15 + X4C10 - 2 X10C5 \leq 1$
c 608 : $X1C15 + X4C11 - 2 X10C4 \leq 1$
c 609 : $X1C15 + X4C12 - 2 X10C3 \leq 1$
c 610 : $X1C15 + X4C13 - 2 X10C2 \leq 1$
c 611 : $X1C15 + X4C14 - 2 X10C1 \leq 1$
c 612 : $X2C1 + X4C2 - 2 X11C3 \leq 1$
c 613 : $X2C1 + X4C3 - 2 X11C2 \leq 1$
c 614 : $X2C1 + X4C4 - 2 X11C5 \leq 1$
c 615 : $X2C1 + X4C5 - 2 X11C4 \leq 1$
c 616 : $X2C1 + X4C6 - 2 X11C7 \leq 1$
c 617 : $X2C1 + X4C7 - 2 X11C6 \leq 1$
c 618 : $X2C1 + X4C8 - 2 X11C9 \leq 1$
c 619 : $X2C1 + X4C9 - 2 X11C8 \leq 1$
c 620 : $X2C1 + X4C10 - 2 X11C11 \leq 1$
c 621 : $X2C1 + X4C11 - 2 X11C10 \leq 1$
c 622 : $X2C1 + X4C12 - 2 X11C13 \leq 1$
c 623 : $X2C1 + X4C13 - 2 X11C12 \leq 1$
c 624 : $X2C1 + X4C14 - 2 X11C15 \leq 1$
c 625 : $X2C1 + X4C15 - 2 X11C14 \leq 1$
c 626 : $X2C2 + X4C1 - 2 X11C3 \leq 1$
c 627 : $X2C2 + X4C3 - 2 X11C1 \leq 1$
c 628 : $X2C2 + X4C4 - 2 X11C6 \leq 1$
c 629 : $X2C2 + X4C5 - 2 X11C7 \leq 1$
c 630 : $X2C2 + X4C6 - 2 X11C4 \leq 1$
c 631 : $X2C2 + X4C7 - 2 X11C5 \leq 1$
c 632 : $X2C2 + X4C8 - 2 X11C10 \leq 1$
c 633 : $X2C2 + X4C9 - 2 X11C11 \leq 1$
c 634 : $X2C2 + X4C10 - 2 X11C8 \leq 1$
c 635 : $X2C2 + X4C11 - 2 X11C9 \leq 1$
c 636 : $X2C2 + X4C12 - 2 X11C14 \leq 1$
c 637 : $X2C2 + X4C13 - 2 X11C15 \leq 1$
c 638 : $X2C2 + X4C14 - 2 X11C12 \leq 1$
c 639 : $X2C2 + X4C15 - 2 X11C13 \leq 1$
c 640 : $X2C3 + X4C1 - 2 X11C2 \leq 1$
c 641 : $X2C3 + X4C2 - 2 X11C1 \leq 1$
c 642 : $X2C3 + X4C4 - 2 X11C7 \leq 1$
c 643 : $X2C3 + X4C5 - 2 X11C6 \leq 1$
c 644 : $X2C3 + X4C6 - 2 X11C5 \leq 1$
c 645 : $X2C3 + X4C7 - 2 X11C4 \leq 1$
c 646 : $X2C3 + X4C8 - 2 X11C11 \leq 1$

c 647: $X2C3 + X4C9 - 2 X11C10 \leq 1$
c 648: $X2C3 + X4C10 - 2 X11C9 \leq 1$
c 649: $X2C3 + X4C11 - 2 X11C8 \leq 1$
c 650: $X2C3 + X4C12 - 2 X11C15 \leq 1$
c 651: $X2C3 + X4C13 - 2 X11C14 \leq 1$
c 652: $X2C3 + X4C14 - 2 X11C13 \leq 1$
c 653: $X2C3 + X4C15 - 2 X11C12 \leq 1$
c 654: $X2C4 + X4C1 - 2 X11C5 \leq 1$
c 655: $X2C4 + X4C2 - 2 X11C6 \leq 1$
c 656: $X2C4 + X4C3 - 2 X11C7 \leq 1$
c 657: $X2C4 + X4C5 - 2 X11C1 \leq 1$
c 658: $X2C4 + X4C6 - 2 X11C2 \leq 1$
c 659: $X2C4 + X4C7 - 2 X11C3 \leq 1$
c 660: $X2C4 + X4C8 - 2 X11C12 \leq 1$
c 661: $X2C4 + X4C9 - 2 X11C13 \leq 1$
c 662: $X2C4 + X4C10 - 2 X11C14 \leq 1$
c 663: $X2C4 + X4C11 - 2 X11C15 \leq 1$
c 664: $X2C4 + X4C12 - 2 X11C8 \leq 1$
c 665: $X2C4 + X4C13 - 2 X11C9 \leq 1$
c 666: $X2C4 + X4C14 - 2 X11C10 \leq 1$
c 667: $X2C4 + X4C15 - 2 X11C11 \leq 1$
c 668: $X2C5 + X4C1 - 2 X11C4 \leq 1$
c 669: $X2C5 + X4C2 - 2 X11C7 \leq 1$
c 670: $X2C5 + X4C3 - 2 X11C6 \leq 1$
c 671: $X2C5 + X4C4 - 2 X11C1 \leq 1$
c 672: $X2C5 + X4C6 - 2 X11C3 \leq 1$
c 673: $X2C5 + X4C7 - 2 X11C2 \leq 1$
c 674: $X2C5 + X4C8 - 2 X11C13 \leq 1$
c 675: $X2C5 + X4C9 - 2 X11C12 \leq 1$
c 676: $X2C5 + X4C10 - 2 X11C15 \leq 1$
c 677: $X2C5 + X4C11 - 2 X11C14 \leq 1$
c 678: $X2C5 + X4C12 - 2 X11C9 \leq 1$
c 679: $X2C5 + X4C13 - 2 X11C8 \leq 1$
c 680: $X2C5 + X4C14 - 2 X11C11 \leq 1$
c 681: $X2C5 + X4C15 - 2 X11C10 \leq 1$
c 682: $X2C6 + X4C1 - 2 X11C7 \leq 1$
c 683: $X2C6 + X4C2 - 2 X11C4 \leq 1$
c 684: $X2C6 + X4C3 - 2 X11C5 \leq 1$
c 685: $X2C6 + X4C4 - 2 X11C2 \leq 1$
c 686: $X2C6 + X4C5 - 2 X11C3 \leq 1$
c 687: $X2C6 + X4C7 - 2 X11C1 \leq 1$
c 688: $X2C6 + X4C8 - 2 X11C14 \leq 1$
c 689: $X2C6 + X4C9 - 2 X11C15 \leq 1$
c 690: $X2C6 + X4C10 - 2 X11C12 \leq 1$
c 691: $X2C6 + X4C11 - 2 X11C13 \leq 1$
c 692: $X2C6 + X4C12 - 2 X11C10 \leq 1$
c 693: $X2C6 + X4C13 - 2 X11C11 \leq 1$
c 694: $X2C6 + X4C14 - 2 X11C8 \leq 1$
c 695: $X2C6 + X4C15 - 2 X11C9 \leq 1$
c 696: $X2C7 + X4C1 - 2 X11C6 \leq 1$
c 697: $X2C7 + X4C2 - 2 X11C5 \leq 1$
c 698: $X2C7 + X4C3 - 2 X11C4 \leq 1$
c 699: $X2C7 + X4C4 - 2 X11C3 \leq 1$
c 700: $X2C7 + X4C5 - 2 X11C2 \leq 1$
c 701: $X2C7 + X4C6 - 2 X11C1 \leq 1$
c 702: $X2C7 + X4C8 - 2 X11C15 \leq 1$

c 703 : $X2C7 + X4C9 - 2 X11C14 \leq 1$
 c 704 : $X2C7 + X4C10 - 2 X11C13 \leq 1$
 c 705 : $X2C7 + X4C11 - 2 X11C12 \leq 1$
 c 706 : $X2C7 + X4C12 - 2 X11C11 \leq 1$
 c 707 : $X2C7 + X4C13 - 2 X11C10 \leq 1$
 c 708 : $X2C7 + X4C14 - 2 X11C9 \leq 1$
 c 709 : $X2C7 + X4C15 - 2 X11C8 \leq 1$
 c 710 : $X2C8 + X4C1 - 2 X11C9 \leq 1$
 c 711 : $X2C8 + X4C2 - 2 X11C10 \leq 1$
 c 712 : $X2C8 + X4C3 - 2 X11C11 \leq 1$
 c 713 : $X2C8 + X4C4 - 2 X11C12 \leq 1$
 c 714 : $X2C8 + X4C5 - 2 X11C13 \leq 1$
 c 715 : $X2C8 + X4C6 - 2 X11C14 \leq 1$
 c 716 : $X2C8 + X4C7 - 2 X11C15 \leq 1$
 c 717 : $X2C8 + X4C9 - 2 X11C1 \leq 1$
 c 718 : $X2C8 + X4C10 - 2 X11C2 \leq 1$
 c 719 : $X2C8 + X4C11 - 2 X11C3 \leq 1$
 c 720 : $X2C8 + X4C12 - 2 X11C4 \leq 1$
 c 721 : $X2C8 + X4C13 - 2 X11C5 \leq 1$
 c 722 : $X2C8 + X4C14 - 2 X11C6 \leq 1$
 c 723 : $X2C8 + X4C15 - 2 X11C7 \leq 1$
 c 724 : $X2C9 + X4C1 - 2 X11C8 \leq 1$
 c 725 : $X2C9 + X4C2 - 2 X11C11 \leq 1$
 c 726 : $X2C9 + X4C3 - 2 X11C10 \leq 1$
 c 727 : $X2C9 + X4C4 - 2 X11C13 \leq 1$
 c 728 : $X2C9 + X4C5 - 2 X11C12 \leq 1$
 c 729 : $X2C9 + X4C6 - 2 X11C15 \leq 1$
 c 730 : $X2C9 + X4C7 - 2 X11C14 \leq 1$
 c 731 : $X2C9 + X4C8 - 2 X11C1 \leq 1$
 c 732 : $X2C9 + X4C10 - 2 X11C3 \leq 1$
 c 733 : $X2C9 + X4C11 - 2 X11C2 \leq 1$
 c 734 : $X2C9 + X4C12 - 2 X11C5 \leq 1$
 c 735 : $X2C9 + X4C13 - 2 X11C4 \leq 1$
 c 736 : $X2C9 + X4C14 - 2 X11C7 \leq 1$
 c 737 : $X2C9 + X4C15 - 2 X11C6 \leq 1$
 c 738 : $X2C10 + X4C1 - 2 X11C11 \leq 1$
 c 739 : $X2C10 + X4C2 - 2 X11C8 \leq 1$
 c 740 : $X2C10 + X4C3 - 2 X11C9 \leq 1$
 c 741 : $X2C10 + X4C4 - 2 X11C14 \leq 1$
 c 742 : $X2C10 + X4C5 - 2 X11C15 \leq 1$
 c 743 : $X2C10 + X4C6 - 2 X11C12 \leq 1$
 c 744 : $X2C10 + X4C7 - 2 X11C13 \leq 1$
 c 745 : $X2C10 + X4C8 - 2 X11C2 \leq 1$
 c 746 : $X2C10 + X4C9 - 2 X11C3 \leq 1$
 c 747 : $X2C10 + X4C11 - 2 X11C1 \leq 1$
 c 748 : $X2C10 + X4C12 - 2 X11C6 \leq 1$
 c 749 : $X2C10 + X4C13 - 2 X11C7 \leq 1$
 c 750 : $X2C10 + X4C14 - 2 X11C4 \leq 1$
 c 751 : $X2C10 + X4C15 - 2 X11C5 \leq 1$
 c 752 : $X2C11 + X4C1 - 2 X11C10 \leq 1$
 c 753 : $X2C11 + X4C2 - 2 X11C9 \leq 1$
 c 754 : $X2C11 + X4C3 - 2 X11C8 \leq 1$
 c 755 : $X2C11 + X4C4 - 2 X11C15 \leq 1$
 c 756 : $X2C11 + X4C5 - 2 X11C14 \leq 1$
 c 757 : $X2C11 + X4C6 - 2 X11C13 \leq 1$
 c 758 : $X2C11 + X4C7 - 2 X11C12 \leq 1$

c 759: $X2C11 + X4C8 - 2 X11C3 \leq 1$
c 760: $X2C11 + X4C9 - 2 X11C2 \leq 1$
c 761: $X2C11 + X4C10 - 2 X11C1 \leq 1$
c 762: $X2C11 + X4C12 - 2 X11C7 \leq 1$
c 763: $X2C11 + X4C13 - 2 X11C6 \leq 1$
c 764: $X2C11 + X4C14 - 2 X11C5 \leq 1$
c 765: $X2C11 + X4C15 - 2 X11C4 \leq 1$
c 766: $X2C12 + X4C1 - 2 X11C13 \leq 1$
c 767: $X2C12 + X4C2 - 2 X11C14 \leq 1$
c 768: $X2C12 + X4C3 - 2 X11C15 \leq 1$
c 769: $X2C12 + X4C4 - 2 X11C8 \leq 1$
c 770: $X2C12 + X4C5 - 2 X11C9 \leq 1$
c 771: $X2C12 + X4C6 - 2 X11C10 \leq 1$
c 772: $X2C12 + X4C7 - 2 X11C11 \leq 1$
c 773: $X2C12 + X4C8 - 2 X11C4 \leq 1$
c 774: $X2C12 + X4C9 - 2 X11C5 \leq 1$
c 775: $X2C12 + X4C10 - 2 X11C6 \leq 1$
c 776: $X2C12 + X4C11 - 2 X11C7 \leq 1$
c 777: $X2C12 + X4C13 - 2 X11C1 \leq 1$
c 778: $X2C12 + X4C14 - 2 X11C2 \leq 1$
c 779: $X2C12 + X4C15 - 2 X11C3 \leq 1$
c 780: $X2C13 + X4C1 - 2 X11C12 \leq 1$
c 781: $X2C13 + X4C2 - 2 X11C15 \leq 1$
c 782: $X2C13 + X4C3 - 2 X11C14 \leq 1$
c 783: $X2C13 + X4C4 - 2 X11C9 \leq 1$
c 784: $X2C13 + X4C5 - 2 X11C8 \leq 1$
c 785: $X2C13 + X4C6 - 2 X11C11 \leq 1$
c 786: $X2C13 + X4C7 - 2 X11C10 \leq 1$
c 787: $X2C13 + X4C8 - 2 X11C5 \leq 1$
c 788: $X2C13 + X4C9 - 2 X11C4 \leq 1$
c 789: $X2C13 + X4C10 - 2 X11C7 \leq 1$
c 790: $X2C13 + X4C11 - 2 X11C6 \leq 1$
c 791: $X2C13 + X4C12 - 2 X11C1 \leq 1$
c 792: $X2C13 + X4C14 - 2 X11C3 \leq 1$
c 793: $X2C13 + X4C15 - 2 X11C2 \leq 1$
c 794: $X2C14 + X4C1 - 2 X11C15 \leq 1$
c 795: $X2C14 + X4C2 - 2 X11C12 \leq 1$
c 796: $X2C14 + X4C3 - 2 X11C13 \leq 1$
c 797: $X2C14 + X4C4 - 2 X11C10 \leq 1$
c 798: $X2C14 + X4C5 - 2 X11C11 \leq 1$
c 799: $X2C14 + X4C6 - 2 X11C8 \leq 1$
c 800: $X2C14 + X4C7 - 2 X11C9 \leq 1$
c 801: $X2C14 + X4C8 - 2 X11C6 \leq 1$
c 802: $X2C14 + X4C9 - 2 X11C7 \leq 1$
c 803: $X2C14 + X4C10 - 2 X11C4 \leq 1$
c 804: $X2C14 + X4C11 - 2 X11C5 \leq 1$
c 805: $X2C14 + X4C12 - 2 X11C2 \leq 1$
c 806: $X2C14 + X4C13 - 2 X11C3 \leq 1$
c 807: $X2C14 + X4C15 - 2 X11C1 \leq 1$
c 808: $X2C15 + X4C1 - 2 X11C14 \leq 1$
c 809: $X2C15 + X4C2 - 2 X11C13 \leq 1$
c 810: $X2C15 + X4C3 - 2 X11C12 \leq 1$
c 811: $X2C15 + X4C4 - 2 X11C11 \leq 1$
c 812: $X2C15 + X4C5 - 2 X11C10 \leq 1$
c 813: $X2C15 + X4C6 - 2 X11C9 \leq 1$
c 814: $X2C15 + X4C7 - 2 X11C8 \leq 1$

c 815 : $X2C15 + X4C8 - 2 X11C7 \leq 1$
 c 816 : $X2C15 + X4C9 - 2 X11C6 \leq 1$
 c 817 : $X2C15 + X4C10 - 2 X11C5 \leq 1$
 c 818 : $X2C15 + X4C11 - 2 X11C4 \leq 1$
 c 819 : $X2C15 + X4C12 - 2 X11C3 \leq 1$
 c 820 : $X2C15 + X4C13 - 2 X11C2 \leq 1$
 c 821 : $X2C15 + X4C14 - 2 X11C1 \leq 1$

This constraint is the minimum aberration constraint. The design matrix columns 1,2,4,7,8,11,13, and 14 were identified as part of the problem as the columns associated with a minimum aberration design. If X1 through X8 are assigned to columns 1,2,4,7,8,11,13, and 14 (in any order), then the design must be a minimum aberration design. If this constraint is not satisfied, then the design may or may not be an equivalent minimum aberration design. The experimenter could evaluate the final design, but the IP has no other method of identifying the aberration of a design. In minimizing the objective function, the IP will try to set $Y = 0$. The only way to make Y equal zero is by assigning X1-X8 to columns 1,2,4,7,8,11,13, and 14. If the sum of the X_iC_i 's is seven or less, then Y must be set to one to make the total greater than or equal to eight.

c 822 : $8 Y + X1C1 + X1C2 + X1C4 + X1C8 + X1C14 + X1C13 + X1C7 + X1C11 + X2C1 + X2C2 + X2C4 + X2C8 + X2C14 + X2C13 + X2C7 + X2C11 + X3C1 + X3C2 + X3C4 + X3C8 + X3C14 + X3C13 + X3C7 + X3C11 + X4C1 + X4C2 + X4C4 + X4C8 + X4C14 + X4C13 + X4C7 + X4C11 + X5C1 + X5C2 + X5C4 + X5C8 + X5C14 + X5C13 + X5C7 + X5C11 + X6C1 + X6C2 + X6C4 + X6C8 + X6C14 + X6C13 + X6C7 + X6C11 + X7C1 + X7C2 + X7C4 + X7C8 + X7C14 + X7C13 + X7C7 + X7C11 + X8C1 + X8C2 + X8C4 + X8C8 + X8C14 + X8C13 + X8C7 + X8C11 \geq 8$

All 177 variables are 0-1 integer variables

There are 177 0-1 variables in this problem and 822 constraints. For comparison, the 64/63 DOE/RS problem has approximately 4000 variables and 150000 constraints.

This problem was solved quickly and easily. 14,384 evaluations were used to execute the IP formulation. The optimal solution had an objective function value of zero and had the

following variables equal to one: X1C1, X2C2, X3C4, X4C8, X5C7, X6C11, X7C13, X8C14, (and the interaction terms AB, AD, and BD) X9C3, X10C9, X11C10.

This problem falls into the class of those that the CAVE heuristic was proved to solve to optimality. There are only 4 variables used in the interaction terms in the RS. Since $S = 4$, an optimal solution was found by CAVE.

D. Codes Used to Write the 0/1 IP Problem File

The following is the Quick basic 4.5 code used to generate the 64/63 sized DOE/RS problem. The "REM" statements are included to self document the software. This code was slightly modified to create an LP-format file for CPLEX for each of the nine baseline DOE/RS problems.

These modifications involved changing the experimental values defined in lines 10 -360.

This information defines the RS and the size of the experiment.

```
10          REM program to write file to create 0-1 IP problem file for the 64/63 problem
20 s = 6:    REM 2^s is the size of the problem. 2^12-8 design
30 p = 17:   REM p is the number of first-order terms in the RS
40 m = 63:   REM m is equal to the total number of terms in the RS.
50 DIM CI(m): REM Ci(i) is the weight assigned to the ith term of the RS.
60 Cminab = 10: REM Cminab is the wt assigned to selecting a minimum aberration design.
70 DIM CA(m): REM the column assignment of the mth term of the RS.
80 DIM DMA(m, s): REM 0-1 representation of the column assignment A=(1,0,0...0)
85 DIM d(p):  REM D is the set of columns associated with a specified min. aberration design.
90 it = m - p: REM it equals the number of interaction terms.
100 DIM ita(it, 2): REM assumes only 2fi's. Defines interaction terms.
105 DIM qita(it, 2): REM the number of digits in the ita term.
115 d$ = "no": REM d$=no means no minimum aberration set has been defined. Skip sections
involving Y
117 IF d$ = "no" THEN GOTO 200
120 d(1) = 1
130 d(2) = 2
140 d(3) = 4
150 d(4) = 8
155 d(5) = 7
160 d(6) = 12
165 d(7) = 14
200          REM input data
210 FOR i = 1 TO m: REM assigns ci's
215 x = 0
220 IF i < p + 1 THEN x = 100
230 CI(i) = x + i
240 NEXT i
300 REM define 2fi's. ita(x,y) is the xth interaction term and the two values in the y position indicate the two
301 REM first order terms as they appear in the RS. The first term is 1, the second 2, and so on.
310 ita(1, 1) = 1: ita(1, 2) = 2: REM ab
311 ita(2, 1) = 3: ita(2, 2) = 4: REM cd
312 ita(3, 1) = 5: ita(3, 2) = 6: REM ef
313 ita(4, 1) = 7: ita(4, 2) = 8: REM gh
314 ita(5, 1) = 9: ita(5, 2) = 10: REM ij
315 ita(6, 1) = 11: ita(6, 2) = 12: REM kl
316 ita(7, 1) = 13: ita(7, 2) = 14: REM mn
317 ita(8, 1) = 15: ita(8, 2) = 16: REM op
```

```

318 ita(9, 1) = 1: ita(9, 2) = 17:      REM aq
319 ita(10, 1) = 1: ita(10, 2) = 3:     REM ac
320 ita(11, 1) = 2: ita(11, 2) = 4:     REM bd
321 ita(12, 1) = 3: ita(12, 2) = 5:     REM ce
322 ita(13, 1) = 4: ita(13, 2) = 6:     REM df
323 ita(14, 1) = 5: ita(14, 2) = 7:     REM eg
324 ita(15, 1) = 6: ita(15, 2) = 8:     REM fh
325 ita(16, 1) = 7: ita(16, 2) = 9:     REM gi
326 ita(17, 1) = 8: ita(17, 2) = 10:    REM hj
327 ita(18, 1) = 9: ita(18, 2) = 11:    REM ik
328 ita(19, 1) = 10: ita(19, 2) = 12: REM jl
329 ita(20, 1) = 11: ita(20, 2) = 13: REM km
330 ita(21, 1) = 12: ita(21, 2) = 14: REM ln
331 ita(22, 1) = 13: ita(22, 2) = 15: REM mo
332 ita(23, 1) = 14: ita(23, 2) = 16: REM np
333 ita(24, 1) = 15: ita(24, 2) = 17: REM oq
334 ita(25, 1) = 1: ita(25, 2) = 4:     REM ad
335 ita(26, 1) = 2: ita(26, 2) = 5:     REM be
336 ita(27, 1) = 3: ita(27, 2) = 6:     REM cf
337 ita(28, 1) = 4: ita(28, 2) = 7:     REM dg
338 ita(29, 1) = 5: ita(29, 2) = 8:     REM eh
339 ita(30, 1) = 6: ita(30, 2) = 9:     REM fi
340 ita(31, 1) = 7: ita(31, 2) = 10:    REM gj
341 ita(32, 1) = 8: ita(32, 2) = 11:    REM hk
342 ita(33, 1) = 9: ita(33, 2) = 12:    REM il
343 ita(34, 1) = 10: ita(34, 2) = 13: REM jm
344 ita(35, 1) = 11: ita(35, 2) = 14: REM kn
345 ita(36, 1) = 12: ita(36, 2) = 15: REM lo
346 ita(37, 1) = 13: ita(37, 2) = 16: REM mp
347 ita(38, 1) = 14: ita(38, 2) = 17: REM nq
348 ita(39, 1) = 1: ita(39, 2) = 5:     REM ae
349 ita(40, 1) = 2: ita(40, 2) = 6:     REM bf
350 ita(41, 1) = 3: ita(41, 2) = 7:     REM cg
351 ita(42, 1) = 4: ita(42, 2) = 8:     REM dh
352 ita(43, 1) = 5: ita(43, 2) = 9:     REM ei
353 ita(44, 1) = 6: ita(44, 2) = 10:    REM fj
354 ita(45, 1) = 7: ita(45, 2) = 11:    REM gk
355 ita(46, 1) = 8: ita(46, 2) = 12:    REM hl

359 FOR i = 1 TO it
360 FOR j = 1 TO 2
370 IF ita(i, j) > 99 THEN qita(i, j) = 3: GOTO 400
380 IF ita(i, j) > 9 THEN qita(i, j) = 2: GOTO 400
390 qita(i, j) = 1
400 NEXT j
410 NEXT i
1200 OPEN "IP64l.lp" FOR OUTPUT AS #1: REM file to store ip format
1210 PRINT #1, "Minimize"
1220 PRINT #1, " Obj: ";
1230 FOR i = 1 TO m: REM print objective
1231 IF i < 99 THEN k1 = 2
1232 IF i < 9.5 THEN k1 = 1
1240 PRINT #1, CI(i); " X"; RIGHT$(STR$(i), k1);
1250 IF i < m THEN PRINT #1, " + ";
1260 NEXT i
1265 IF d$ = "no" THEN PRINT #1, " ": GOTO 1280

```

```

1270 PRINT #1, " + "; Cminab; " Y": REM end of objective statement
1280 PRINT #1, "Subject to"
1290 column = 1: REM define first constraint line number
1300 REM 1st constraint Assignment constraint
1310 n = 2 ^ s - 1: REM the number of columns in the design
1320 FOR i = 1 TO m: REM Once for each element in the RS
1322 PRINT #1, "c"; column; ": "; column = column + 1
1325 IF i > 99 THEN k1 = 3: GOTO 1330
1326 IF i > 9 THEN k1 = 2: GOTO 1330
1327 k1 = 1
1330 FOR j = 1 TO n: REM summing over all columns, must be assigned to one.
1341 IF j > 99 THEN k2 = 3: GOTO 1350
1342 IF j > 9 THEN k2 = 2: GOTO 1350
1343 k2 = 1
1350 IF j = n THEN GOTO 1380
1360 PRINT #1, "X" + RIGHT$(STR$(i), k1) + "C" + RIGHT$(STR$(j), k2);
1365 PRINT #1, " + ";
1367 IF j = 1 THEN PRINT #1, " & ";
1370 NEXT j
1380 PRINT #1, "X" + RIGHT$(STR$(i), k1) + "C" + RIGHT$(STR$(j), k2); " = 1"
1390 NEXT i
2000 REM Resolution III constraint
2010 FOR i = 1 TO p: REM Once for each first-order term in the RS
2020 PRINT #1, "c"; column; ": "; column = column + 1
2025 IF i > 99 THEN k1 = 3: GOTO 2030
2026 IF i > 9 THEN k1 = 2: GOTO 2030
2027 k1 = 1
2030 FOR j = 1 TO n: REM summing over all columns, must be assigned to one.
2041 IF j > 99 THEN k2 = 3: GOTO 2050
2042 IF j > 9 THEN k2 = 2: GOTO 2050
2043 k2 = 1
2050 PRINT #1, "X" + RIGHT$(STR$(i), k1) + "C" + RIGHT$(STR$(j), k2);
2060 IF j < n THEN PRINT #1, " + ";
2065 IF j = 1 THEN PRINT #1, " & "; : REM bug fix
2070 NEXT j
2080 PRINT #1, " <= 1"
2090 NEXT i
3000 REM Confounding constraint
3010 FOR i = 1 TO m: REM Once for each first-order term in the RS
3025 IF i > 99 THEN k1 = 3: GOTO 3030
3026 IF i > 9 THEN k1 = 2: GOTO 3030
3027 k1 = 1
3030 FOR j = 1 TO n: REM summing over all columns, must be assigned to one.
3041 IF j > 99 THEN k2 = 3: GOTO 3045
3042 IF j > 9 THEN k2 = 2: GOTO 3045
3043 k2 = 1
3045 PRINT #1, "c"; column; ": "; column = column + 1
3050 PRINT #1, "-"; n; "X" + RIGHT$(STR$(i), k1); " + "; n; "X" + RIGHT$(STR$(i), k1) + "C" +
RIGHT$(STR$(j), k2); " + ";
3060 FOR kz = 1 TO n: REM sum over all assigned to jth column
3070 IF kz > 99 THEN k3 = 3: GOTO 3100
3080 IF kz > 9 THEN k3 = 2: GOTO 3100
3090 k3 = 1
3100 PRINT #1, "X" + RIGHT$(STR$(kz), k3) + "C" + RIGHT$(STR$(j), k2);
3110 IF kz < n THEN PRINT #1, " + ";
3140 NEXT kz

```

```

3150 PRINT #1, " <= "; n + 1
3200 NEXT j
3300 NEXT i
4000                                REM Interaction constraints: using base two addition on the vector
representation
4010                                REM will result in a vector representation of the assignment of the 2fi
term.
4015 FOR h = p + 1 TO m: REM once for each interaction term in RS
4020 IF h > 99 THEN qh = 3: GOTO 4100
4030 IF h > 9 THEN qh = 2: GOTO 4100
4040 qh = 1
4100 FOR i6 = 0 TO 1
4105 FOR i5 = 0 TO 1
4108 FOR i4 = 0 TO 1
4110 FOR i3 = 0 TO 1
4120 FOR i2 = 0 TO 1
4140 FOR i1 = 0 TO 1
4170 i = i1 + 2 * i2 + 4 * i3 + 8 * i4 + 16 * i5 + 32 * i6
4175 IF i = 0 THEN GOTO 4470: REM 0,0,0 is not a column
4180 IF i > 99 THEN q1 = 3: GOTO 4200
4185 IF i > 9 THEN q1 = 2: GOTO 4200
4190 q1 = 1
4200 FOR j6 = 0 TO 1
4205 FOR j5 = 0 TO 1
4208 FOR j4 = 0 TO 1
4210 FOR j3 = 0 TO 1
4220 FOR j2 = 0 TO 1
4240 FOR j1 = 0 TO 1
4270 j = j1 + 2 * j2 + 4 * j3 + 8 * j4 + 16 * j5 + 32 * j6
4277 IF i = j THEN GOTO 4440: REM not a possible assignment
4275 IF j = 0 THEN GOTO 4440: REM 0,0,0 is not a column
4280 IF j > 99 THEN q2 = 3: GOTO 4300
4285 IF j > 9 THEN q2 = 2: GOTO 4300
4290 q2 = 1
4300 REM calculating k1,k2,k3
4305 k6 = j6 + i6: IF k6 > 1.5 THEN k6 = k6 - 2
4308 k5 = j5 + i5: IF k5 > 1.5 THEN k5 = k5 - 2
4310 k4 = j4 + i4: IF k4 > 1.5 THEN k4 = k4 - 2
4315 k1 = j1 + i1: IF k1 > 1.5 THEN k1 = k1 - 2
4320 k2 = j2 + i2: IF k2 > 1.5 THEN k2 = k2 - 2
4330 k3 = j3 + i3: IF k3 > 1.5 THEN k3 = k3 - 2
4340 kt = k1 + 2 * k2 + 4 * k3 + 8 * k4 + 16 * k5 + 32 * k6
4350 IF kt > 99 THEN q3 = 3: GOTO 4395
4385 IF kt > 9 THEN q3 = 2: GOTO 4395
4390 q3 = 1
4395 x = h - p
4400 PRINT #1, "c"; column; ": "; : column = column + 1
4410 PRINT #1, "X" + RIGHT$(STR$(ita(x, 1)), q1a(x, 1)) + "C" + RIGHT$(STR$(i), q1); " + ";
4420 PRINT #1, "X" + RIGHT$(STR$(ita(x, 2)), q1a(x, 2)) + "C" + RIGHT$(STR$(j), q2); " - ";
4430 PRINT #1, "2 "; "X" + RIGHT$(STR$(h), qh) + "C" + RIGHT$(STR$(kt), q3); " <= "; " 1"
4440 NEXT j1
4450 NEXT j2
4460 NEXT j3
4463 NEXT j4
4465 NEXT j5
4466 NEXT j6

```

```

4470 NEXT i1
4480 NEXT i2
4490 NEXT i3
4493 NEXT i4
4494 NEXT i5
4495 NEXT i6
4500 NEXT h
5000 REM minimum aberration constraint
5005 IF d$ = "no" THEN GOTO 6000
5010 PRINT #1, "c"; column; ": "; : column = column + 1
5020 PRINT #1, p; "Y "; " + ";
5030 FOR i = 1 TO p: REM p first-order terms
5032 IF i > 99 THEN k1 = 3: GOTO 5040
5035 IF i > 9 THEN k1 = 2: GOTO 5040
5038 k1 = 1
5040 FOR j = 1 TO p: REM p columns in d
5042 IF d(j) > 99 THEN k2 = 3: GOTO 5050
5045 IF d(j) > 9 THEN k2 = 2: GOTO 5050
5048 k2 = 1
5050 PRINT #1, "X" + RIGHT$(STR$(i), k1) + "C" + RIGHT$(STR$(d(j)), k2);
5060 IF i + j < 2 * p THEN PRINT #1, " + ";
5070 NEXT j
5080 NEXT i
5100 PRINT #1, " >= "; p
6000 REM print bounds on variables and integer statement
6005 REM PRINT #1, "Bounds"
6010 REM FOR i = 1 TO m
6013 REM IF i > 99 THEN k1 = 3: GOTO 6020
6015 REM IF i > 9 THEN k1 = 2: GOTO 6020
6018 REM k1 = 1
6020 REM FOR j = 1 TO n
6023 REM IF j > 99 THEN k2 = 3: GOTO 6030
6025 REM IF j > 9 THEN k2 = 2: GOTO 6030
6028 REM k2 = 1
6030 REM PRINT #1, "0 <= "; "X" + RIGHT$(STR$(i), k1) + "C" + RIGHT$(STR$(j), k2); " <= 1"
6040 REM NEXT j
6050 REM NEXT i
6120 REM FOR i = 1 TO m
6123 REM IF i > 99 THEN k1 = 3: GOTO 6130
6125 REM IF i > 9 THEN k1 = 2: GOTO 6130
6128 REM k1 = 1
6130 REM PRINT #1, "0 <= "; "X" + RIGHT$(STR$(i), k1); " <= 1"
6150 REM NEXT i
6190 REM IF d$ = "no" THEN GOTO 6300
6200 REM PRINT #1, "0 <= Y <= 1"
6300 PRINT #1, "Integer"
6310 FOR i = 1 TO m
6313 IF i > 99 THEN k1 = 3: GOTO 6320
6315 IF i > 9 THEN k1 = 2: GOTO 6320
6318 k1 = 1
6320 FOR j = 1 TO n
6323 IF j > 99 THEN k2 = 3: GOTO 6330
6325 IF j > 9 THEN k2 = 2: GOTO 6330
6328 k2 = 1
6330 PRINT #1, "X" + RIGHT$(STR$(i), k1) + "C" + RIGHT$(STR$(j), k2),
6340 NEXT j

```

```

6345 PRINT #1, " "
6350 NEXT i
6420 FOR i = 1 TO m
6423 IF i > 99 THEN k1 = 3: GOTO 6430
6425 IF i > 9 THEN k1 = 2: GOTO 6430
6428 k1 = 1
6430 PRINT #1, "X" + RIGHT$(STR$(i), k1),
6450 NEXT i
6500 IF d$ = "no" THEN PRINT #1, " ": GOTO 6600
6520 PRINT #1, "Y"
6600 PRINT #1, "END"

9000 CLOSE

```

The following formulation was produced by the code just presented (the inputs, the variables defined in lines 20-360, were changed to reflect the specific DOE/RS problem) The remarks section of each line identifies the variable to aid in changing the inputs to solve new DOE/RS problems.

The code was written to generate problem files that CPLEX would read as an input. Construction of such large problem files by hand would have been inefficient and could have resulted in numerous typographical errors. This program was written in QuickBasic 4.5. The program was modified and then compiled for each of the nine baseline DOE/RS problems.

In order to provide the reader with several examples,, the 16/12 DOE/RS problem is presented rather than continuing with the 64/63 example. Since the 16/12 problem is the smallest of the nine baseline DOE/RS problems, this formulation has the fewest constraints. When more than 15 of a given constraint type exist, the first ten and the last five are specifically listed.

Minimize

Obj: 101 X1 + 102 X2 + 103 X3 + 104 X4 + 105 X5 + 106 X6 + 107 X7 + 8
X8 + 9 X9 + 10 X10 + 11 X11 + 12 X12 + 10 Y

Subject to

c 1 : X1C1 + X1C2 + X1C3 + X1C4 + X1C5 + X1C6 + X1C7 + X1C8 + X1C9 +
X1C10 + X1C11 + X1C12 + X1C13 + X1C14 + X1C15 = 1

c 2 : X2C1 + X2C2 + X2C3 + X2C4 + X2C5 + X2C6 + X2C7 + X2C8 + X2C9 +
X2C10 + X2C11 + X2C12 + X2C13 + X2C14 + X2C15 = 1

c 3 : X3C1 + X3C2 + X3C3 + X3C4 + X3C5 + X3C6 + X3C7 + X3C8 + X3C9 +

$X3C10 + X3C11 + X3C12 + X3C13 + X3C14 + X3C15 = 1$
c 4 : $X4C1 + X4C2 + X4C3 + X4C4 + X4C5 + X4C6 + X4C7 + X4C8 + X4C9 + X4C10 + X4C11 + X4C12 + X4C13 + X4C14 + X4C15 = 1$
c 5 : $X5C1 + X5C2 + X5C3 + X5C4 + X5C5 + X5C6 + X5C7 + X5C8 + X5C9 + X5C10 + X5C11 + X5C12 + X5C13 + X5C14 + X5C15 = 1$
c 6 : $X6C1 + X6C2 + X6C3 + X6C4 + X6C5 + X6C6 + X6C7 + X6C8 + X6C9 + X6C10 + X6C11 + X6C12 + X6C13 + X6C14 + X6C15 = 1$
c 7 : $X7C1 + X7C2 + X7C3 + X7C4 + X7C5 + X7C6 + X7C7 + X7C8 + X7C9 + X7C10 + X7C11 + X7C12 + X7C13 + X7C14 + X7C15 = 1$
c 8 : $X8C1 + X8C2 + X8C3 + X8C4 + X8C5 + X8C6 + X8C7 + X8C8 + X8C9 + X8C10 + X8C11 + X8C12 + X8C13 + X8C14 + X8C15 = 1$
c 9 : $X9C1 + X9C2 + X9C3 + X9C4 + X9C5 + X9C6 + X9C7 + X9C8 + X9C9 + X9C10 + X9C11 + X9C12 + X9C13 + X9C14 + X9C15 = 1$
c 10 : $X10C1 + X10C2 + X10C3 + X10C4 + X10C5 + X10C6 + X10C7 + X10C8 + X10C9 + X10C10 + X10C11 + X10C12 + X10C13 + X10C14 + X10C15 = 1$
c 11 : $X11C1 + X11C2 + X11C3 + X11C4 + X11C5 + X11C6 + X11C7 + X11C8 + X11C9 + X11C10 + X11C11 + X11C12 + X11C13 + X11C14 + X11C15 = 1$
c 12 : $X12C1 + X12C2 + X12C3 + X12C4 + X12C5 + X12C6 + X12C7 + X12C8 + X12C9 + X12C10 + X12C11 + X12C12 + X12C13 + X12C14 + X12C15 = 1$
c 13 : $X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 \leq 1$
c 14 : $X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 \leq 1$
c 15 : $X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 \leq 1$
c 16 : $X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 \leq 1$
c 17 : $X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 \leq 1$
c 18 : $X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 \leq 1$
c 19 : $X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 \leq 1$
c 20 : $X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 \leq 1$
c 21 : $X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 \leq 1$
c 22 : $X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 \leq 1$
c 23 : $X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 \leq 1$
c 24 : $X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 \leq 1$
c 25 : $X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 \leq 1$
c 26 : $X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 \leq 1$
c 27 : $X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 \leq 1$
c 28 : $-15 X1 + 15 X1C1 + X1C1 + X2C1 + X3C1 + X4C1 + X5C1 + X6C1 + X7C1 + X8C1 + X9C1 + X10C1 + X11C1 + X12C1 + X13C1 + X14C1 + X15C1 \leq 16$
c 29 : $-15 X1 + 15 X1C2 + X1C2 + X2C2 + X3C2 + X4C2 + X5C2 + X6C2 + X7C2 + X8C2 + X9C2 + X10C2 + X11C2 + X12C2 + X13C2 + X14C2 + X15C2 \leq 16$
c 30 : $-15 X1 + 15 X1C3 + X1C3 + X2C3 + X3C3 + X4C3 + X5C3 + X6C3 + X7C3 + X8C3 + X9C3 + X10C3 + X11C3 + X12C3 + X13C3 + X14C3 + X15C3 \leq 16$
c 31 : $-15 X1 + 15 X1C4 + X1C4 + X2C4 + X3C4 + X4C4 + X5C4 + X6C4 + X7C4 + X8C4 + X9C4 + X10C4 + X11C4 + X12C4 + X13C4 + X14C4 + X15C4 \leq 16$
c 32 : $-15 X1 + 15 X1C5 + X1C5 + X2C5 + X3C5 + X4C5 + X5C5 + X6C5 + X7C5 + X8C5 + X9C5 + X10C5 + X11C5 + X12C5 + X13C5 + X14C5 + X15C5 \leq 16$
c 33 : $-15 X1 + 15 X1C6 + X1C6 + X2C6 + X3C6 + X4C6 + X5C6 + X6C6 + X7C6 + X8C6 + X9C6 + X10C6 + X11C6 + X12C6 + X13C6 + X14C6 + X15C6 \leq 16$
c 34 : $-15 X1 + 15 X1C7 + X1C7 + X2C7 + X3C7 + X4C7 + X5C7 + X6C7 + X7C7 + X8C7 + X9C7 + X10C7 + X11C7 + X12C7 + X13C7 + X14C7 + X15C7 \leq 16$
c 35 : $-15 X1 + 15 X1C8 + X1C8 + X2C8 + X3C8 + X4C8 + X5C8 + X6C8 + X7C8 + X8C8 + X9C8 + X10C8 + X11C8 + X12C8 + X13C8 + X14C8 + X15C8 \leq 16$
c 36 : $-15 X1 + 15 X1C9 + X1C9 + X2C9 + X3C9 + X4C9 + X5C9 + X6C9 + X7C9 + X8C9 + X9C9 + X10C9 + X11C9 + X12C9 + X13C9 + X14C9 + X15C9 \leq 16$
c 37 : $-15 X1 + 15 X1C10 + X1C10 + X2C10 + X3C10 + X4C10 + X5C10 + X6C10 + X7C10 + X8C10 + X9C10 + X10C10 + X11C10 + X12C10 + X13C10 + X14C10 + X15C10 \leq 16$
c 38 : $-15 X1 + 15 X1C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 \leq 16$
c 39
c 203 : $-15 X12 + 15 X12C11 + X1C11 + X2C11 + X3C11 + X4C11 + X5C11 + X6C11 + X7C11 + X8C11 + X9C11 + X10C11 + X11C11 + X12C11 + X13C11 + X14C11 + X15C11 \leq 16$
c 204 : $-15 X12 + 15 X12C12 + X1C12 + X2C12 + X3C12 + X4C12 + X5C12 + X6C12 + X7C12 + X8C12 + X9C12 + X10C12 + X11C12 + X12C12 + X13C12 +$

$X14C12 + X15C12 \leq 16$
 $c\ 205 : -15 X12 + 15 X12C13 + X1C13 + X2C13 + X3C13 + X4C13 + X5C13 + X6C13 + X7C13 + X8C13 + X9C13 + X10C13 + X11C13 + X12C13 + X13C13 + X14C13 + X15C13 \leq 16$
 $c\ 206 : -15 X12 + 15 X12C14 + X1C14 + X2C14 + X3C14 + X4C14 + X5C14 + X6C14 + X7C14 + X8C14 + X9C14 + X10C14 + X11C14 + X12C14 + X13C14 + X14C14 + X15C14 \leq 16$
 $c\ 207 : -15 X12 + 15 X12C15 + X1C15 + X2C15 + X3C15 + X4C15 + X5C15 + X6C15 + X7C15 + X8C15 + X9C15 + X10C15 + X11C15 + X12C15 + X13C15 + X14C15 + X15C15 \leq 16$
 $c\ 208 : X1C1 + X2C2 - 2 X8C3 \leq 1$
 $c\ 209 : X1C1 + X2C3 - 2 X8C2 \leq 1$
 $c\ 210 : X1C1 + X2C4 - 2 X8C5 \leq 1$
 $c\ 211 : X1C1 + X2C5 - 2 X8C4 \leq 1$
 $c\ 212 : X1C1 + X2C6 - 2 X8C7 \leq 1$
 $c\ 213 : X1C1 + X2C7 - 2 X8C6 \leq 1$
 $c\ 214 : X1C1 + X2C8 - 2 X8C9 \leq 1$
 $c\ 215 : X1C1 + X2C9 - 2 X8C8 \leq 1$
 $c\ 216 : X1C1 + X2C10 - 2 X8C11 \leq 1$
 $c\ 217 : X1C1 + X2C11 - 2 X8C10 \leq 1$
 $c\ 218 : \dots$
 $c\ 1253 : X2C15 + X7C10 - 2 X12C5 \leq 1$
 $c\ 1254 : X2C15 + X7C11 - 2 X12C4 \leq 1$
 $c\ 1255 : X2C15 + X7C12 - 2 X12C3 \leq 1$
 $c\ 1256 : X2C15 + X7C13 - 2 X12C2 \leq 1$
 $c\ 1257 : X2C15 + X7C14 - 2 X12C1 \leq 1$
 $c\ 1258 : 7 Y + X1C1 + X1C2 + X1C4 + X1C8 + X1C7 + X1C12 + X1C14 + X2C1 + X2C2 + X2C4 + X2C8 + X2C7 + X2C12 + X2C14 + X3C1 + X3C2 + X3C4 + X3C8 + X3C7 + X3C12 + X3C14 + X4C1 + X4C2 + X4C4 + X4C8 + X4C7 + X4C12 + X4C14 + X5C1 + X5C2 + X5C4 + X5C8 + X5C7 + X5C12 + X5C14 + X6C1 + X6C2 + X6C4 + X6C8 + X6C7 + X6C12 + X6C14 + X7C1 + X7C2 + X7C4 + X7C8 + X7C7 + X7C12 + X7C14 \geq 7$

Integer

X1C1	X1C2	X1C3	X1C4	X1C5	X1C6	X1C7
X1C8	X1C9	X1C10	X1C11	X1C12	X1C13	X1C14
X1C15						
X2C1	X2C2	X2C3	X2C4	X2C5	X2C6	X2C7
X2C8	X2C9	X2C10	X2C11	X2C12	X2C13	X2C14
X2C15						
X3C1	X3C2	X3C3	X3C4	X3C5	X3C6	X3C7
X3C8	X3C9	X3C10	X3C11	X3C12	X3C13	X3C14
X3C15						
X4C1	X4C2	X4C3	X4C4	X4C5	X4C6	X4C7
X4C8	X4C9	X4C10	X4C11	X4C12	X4C13	X4C14
X4C15						
X5C1	X5C2	X5C3	X5C4	X5C5	X5C6	X5C7
X5C8	X5C9	X5C10	X5C11	X5C12	X5C13	X5C14
X5C15						
X6C1	X6C2	X6C3	X6C4	X6C5	X6C6	X6C7
X6C8	X6C9	X6C10	X6C11	X6C12	X6C13	X6C14
X6C15						
X7C1	X7C2	X7C3	X7C4	X7C5	X7C6	X7C7
X7C8	X7C9	X7C10	X7C11	X7C12	X7C13	X7C14
X7C15						
X8C1	X8C2	X8C3	X8C4	X8C5	X8C6	X8C7
X8C8	X8C9	X8C10	X8C11	X8C12	X8C13	X8C14
X8C15						
X9C1	X9C2	X9C3	X9C4	X9C5	X9C6	X9C7
X9C8	X9C9	X9C10	X9C11	X9C12	X9C13	X9C14
X9C15						
X10C1	X10C2	X10C3	X10C4	X10C5	X10C6	X10C7
X10C8	X10C9	X10C10	X10C11	X10C12	X10C13	X10C14
X10C15						
X11C1	X11C2	X11C3	X11C4	X11C5	X11C6	X11C7

X11C8	X11C9	X11C10	X11C11	X11C12	X11C13	X11C14
X11C15						
X12C1	X12C2	X12C3	X12C4	X12C5	X12C6	X12C7
X12C8	X12C9	X12C10	X12C11	X12C12	X12C13	X12C14
X12C15						
X1	X2	X3	X4	X5	X6	X7
X10	X11	X12	Y			
END						

Instruction to generate the IP formulation:

The REM statements in the code were designed to document the code and the purpose of each section and line in the code. The code to write a new IP formulation is created by editing the existing lines in the code that defines variables. Edit the code with the specifics of the experiment to be constructed. Specifically:

10 REM program to write file to create 0-1 IP problem file for the 64/63 problem

Rewrite the REM statement to identify the problem

20 s = 6: REM 2^s is the size of the problem. 2¹²⁻⁸ design

Change S to the appropriate value.

30 p = 17: REM p is the number of first-order terms in the RS

Change P to the appropriate value.

40 m = 63: REM m is equal to the total number of terms in the RS.

Change M to the appropriate value.

60 Cminab = 10: REM Cminab is the wt assigned to selecting a minimum aberration design.

Change Cminab to the appropriate value.

115 d\$ = "no": REM d\$=no means no minimum aberration set has been defined. Skip sections involving Y

Change d\$ if appropriate to "yes". If there is a minimum aberration set D, then define the columns in the following lines:

120 d(1) = 1

130 d(2) = 2

140 d(3) = 4

150 d(4) = 8

155 d(5) = 7

160 d(6) = 12

165 d(7) = 14

210 FOR i = 1 TO m: REM assigns ci's

215 x = 0

220 IF i < p + 1 THEN x = 100

230 CI(i) = x + i

240 NEXT I

This section sets values for the C_i 's. Adjust as needed.

300 REM define 2fi's. ita(x,y) is the x^{th} interaction term and the two values in the y position indicate the two
301 REM first order terms as they appear in the RS. The first term is 1, the second 2, and so on.

310 ita(1, 1) = 1: ita(1, 2) = 2: REM ab

311 ita(2, 1) = 3: ita(2, 2) = 4: REM cd

312 ita(3, 1) = 5: ita(3, 2) = 6: REM ef

.

.

.

353 ita(44, 1) = 6: ita(44, 2) = 10: REM fj

354 ita(45, 1) = 7: ita(45, 2) = 11: REM gk

355 ita(46, 1) = 8: ita(46, 2) = 12: REM hl

Change these lines to define the interaction terms in the RS.

1200 OPEN "IP64l.lp" FOR OUTPUT AS #1: REM file to store ip format

Change "IP64l.lp" to the filename desired using "filename.extension" format.

Save the changes. Compile the code for fast run times if desired. Then run the
program. The IP formulation will be written as a text file to the "filename.extension."

E. Code Used to Implement the CAVE and ICAVE Heuristics

This section presents another of the nine DOE/RS problems. The different examples used throughout the Appendix were selected so as to maximize the diversity of the examples. Providing different examples may help some readers gain insight that they might otherwise miss from seeing the same example problem repetitively. The following is the code for the ICAVE heuristic for the 32/31 DOE/RS problem. Each DOE/RS problem was solved by slightly modifying the ICAVE program. As mentioned later in the instructions to run the ICAVE heuristic, this code can also execute the CAVE heuristic by setting the maximum number of cycles to one (1) in line 3000. The data to describe each experiment is contained in the code between line 10 and line 300. These changes adjusted for the specific RS and the size of the design matrix.

```
10 REM Heuristic for solving the DOE/RS problem using greedy heuristic ICAVE
20 REM by Major Steven Forsythe 5 Nov 99
25 REM
30 n$ = "cave32l.dat": REM
35 OPEN n$ FOR OUTPUT AS #1
40 CLS
50 REM
60 p = 12: REM INPUT "How many first order terms are in the Requirements set?", p
70 m = 31: REM INPUT "how many total terms are there in the RS?", m
80 s = 5: REM INPUT "What is s, where the experiment size is 2^s", s
90 it = m - p: REM it is the number of interaction terms
100 n = 2 ^ s - 1
110 zcur = zbest = ztemp = 100000

200 DIM x(m): REM X(i) is confounded
210 DIM CI(m): REM c is the weight assigned each term in the RS
215 Cminab = 10000
220 DIM rank(p): REM rank is the ranked order of the first-order terms in RS (highest =1)
225 DIM rkval(p): REM rkval is the sum of the weights of first order terms
228 DIM rktmp(p): REM temporary values of rkval
230 DIM xvble$(p): REM xvble is the name of the experimental variable
233 DIM tvector(m, s): REM tvector(m,s) is a vector which defines a temporary column assignment
234 DIM cvector(m, s): REM current vector
235 DIM bvector(m, s): REM records the best solution vector so far
240 DIM ita(it, 2): REM This term will define each interaction term
245 REM DIM d(p, s): REM d is the min aberration set
250 REM define 2f's
260 ita(1, 1) = 1: ita(1, 2) = 2: REM ab
```

```

261 ita(2, 1) = 3: ita(2, 2) = 4: REM cd
262 ita(3, 1) = 5: ita(3, 2) = 6: REM ef
263 ita(4, 1) = 7: ita(4, 2) = 8: REM gh
264 ita(5, 1) = 9: ita(5, 2) = 10: REM ij
270 ita(6, 1) = 11: ita(6, 2) = 12: REM kl
271 ita(7, 1) = 1: ita(7, 2) = 12: REM al
272 ita(8, 1) = 2: ita(8, 2) = 11: REM bk
273 ita(9, 1) = 3: ita(9, 2) = 10: REM cj
274 ita(10, 1) = 4: ita(10, 2) = 9: REM di
275 ita(11, 1) = 5: ita(11, 2) = 8: REM eh
276 ita(12, 1) = 6: ita(12, 2) = 7: REM fg
277 ita(13, 1) = 2: ita(13, 2) = 7: REM be
280 ita(14, 1) = 1: ita(14, 2) = 4: REM ad
282 ita(15, 1) = 2: ita(15, 2) = 5: REM be
283 ita(16, 1) = 3: ita(16, 2) = 6: REM cf
284 ita(17, 1) = 4: ita(17, 2) = 7: REM dg
285 ita(18, 1) = 1: ita(18, 2) = 5: REM ae
286 ita(19, 1) = 2: ita(19, 2) = 6: REM bf

300 REM this is the routine to input the RS and the wts for each term in the RS
310 REM
320 FOR i = 1 TO p
330 xvble$(i) = "X" + STR$(i): REM INPUT xvble$(i)
331 NEXT i
335 FOR i = 1 TO m: REM add the ci's (weights)
341 x = 0
342 IF i < p + 1 THEN x = 100
343 CI(i) = x + i
344 NEXT i
700 REM check input data
710 CLS
720 REM WRITE #1, "Is the RS as follows?"
725 REM WRITE #1, "#", "name", "weight"
730 REM FOR i = 1 TO p
740 REM WRITE #1, i, xvble$(i), CI(i)
750 REM NEXT i
760 REM FOR i = 1 TO it
770 REM WRITE #1, i + p, ita(i, 1), "*", ita(i, 2), CI(i + p)
780 REM NEXT i
800 REM INPUT "Is this correct (Y/N)", y$
810 REM IF y$ = "n" THEN STOP
820 REM IF y$ = "N" THEN STOP
830 REM IF y$ = "Y" THEN GOTO 1000
840 REM IF y$ = "y" THEN GOTO 1000
850 REM GOTO 800
1000 REM calculate the most "important variables" and rank order them
1010 REM mkval is the rank values for the r first order terms
1020 REM this loop calculates the sum of the weights associated with a term
1030 FOR i = 1 TO p
1040 mkval(i) = CI(i)
1050 NEXT i
1070 REM now to add the 2it terms to the mkval
1080 FOR i = 1 TO it
1090 FOR k = 1 TO 2
1100 FOR j = 1 TO p
1110 IF ita(i, k) = j THEN mkval(j) = mkval(j) + CI(p + i)

```

```

1120 NEXT j
1130 NEXT k
1140 NEXT i: REM this adds wts for the 2it to appropriate rnkval's
1150 REM This section creates temporary values to be used to rank order the rnkval
1160 FOR i = 1 TO p
1170 mktmp(i) = rnkval(i)
1180 NEXT i
1200 REM This section rank orders the first order terms and assigns them to rank(r)
1205 temp = 0
1210 FOR i = 1 TO p: REM i is the ith rank order term
1220 FOR j = 1 TO p: REM j is the jth first order term
1225 IF mktmp(j) > temp THEN temp = mktmp(j): rank(i) = j
1230 NEXT j
1240 mktmp(rank(i)) = -1: REM this change removes the assigned term from future consideration
1250 temp = 0
1260 NEXT i
2000 REM prints the rank order of the first order terms
2010 REM WRITE #1, "This is the experimental variables in order of their significance"
2014 REM WRITE #1, "rank", "variable"
2020 REM FOR i = 1 TO p
2030 REM WRITE #1, i, "X", rank(i)
2040 REM NEXT i
2050 REM INPUT "Continue", x
2999 RANDOMIZE
3000 FOR cycle = 1 TO 100: REM This section assigns first order terms to specific columns
3110 FOR i = 1 TO s
3130 tvector(rank(i), i) = 1: REM The most important term is assigned to (1,0,0...) etc.
3135 cvector(rank(i), i) = 1
3140 bvector(rank(i), i) = 1
3150 NEXT i
3200 REM This section must assign the 2fi's
3210 FOR i = 1 TO it
3220 FOR j = 1 TO s
3230 FOR k = 1 TO s
3235 IF ita(i, 1) = rank(j) THEN IF ita(i, 2) = rank(k) THEN GOSUB 10000
3250 NEXT k
3260 NEXT j
3270 NEXT i
3300 REM This section must assign the additional first order terms to columns
3310 FOR i = s + 1 TO p
3320 GOSUB 20000
3325 FOR j1 = 1 TO m
3330 FOR j = 1 TO s
3332 bvector(j1, j) = cvector(j1, j)
3333 NEXT j
3334 NEXT j1
3390 NEXT i
4400 REM INPUT x: REM test
4500 REM CLS
4520 REM WRITE #1, "    The assignment found is as follows:"
4540 REM FOR i = 1 TO p
4560 REM WRITE #1, xvble$(i), "is assigned to column "
4562 REM FOR j = 1 TO s
4564 REM tempcol = tempcol + (2 ^ (j - 1)) * cvector(i, j)
4566 REM NEXT j
4567 REM WRITE #1, tempcol: tempcol = 0

```

```

4570 REM NEXT i
4571 REM FOR i = p + 1 TO m
4572 REM WRITE #1, "interaction term", i
4573 REM FOR j = 1 TO s
4574 REM tempcol = tempcol + (2 ^ (j - 1)) * cvector(i, j)
4576 REM NEXT j
4577 REM WRITE #1, tempcol: tempcol = 0
4578 REM NEXT i
4579 GOSUB 40000
4580 WRITE #1, "Objective Function Value = ", zcur
4600 REM zbest = 1000: zcur = 1000: REM test code
4700 IF zcur < 50 THEN GOSUB 6000
4800 IF zcur < .1 THEN cycle = 100
5000 NEXT cycle
5500 BEEP: STOP
6000 REM prints current solution
6010 WRITE #1, " ZBest is:"
6020 FOR i = 1 TO m
6025 WRITE #1, i
6030 FOR j = 1 TO s
6040 WRITE #1, cvector(i, j)
6050 NEXT j
6060 WRITE #1,
6070 NEXT i
6100 RETURN

7000 BEEP: STOP

10000 REM This subroutine assigns columns to the i th 2fi = j*k and assign vector
10010 FOR l = 1 TO s
10015 tvector(i + p, l) = tvector(rank(j), l) + tvector(rank(k), l)
10020 IF tvector(i + p, l) = 2 THEN tvector(i + p, l) = 0: REM mod 2 addition
10030 cvector(i + p, l) = tvector(i + p, l)
10040 bvector(i + p, l) = tvector(i + p, l)
10080 NEXT l
10200 RETURN

20000 REM This subroutine assigns s+1 to p first order terms; i = term #
20001 zcur = 0: REM reset current solution on first iteration
20003 FOR j = s + 1 TO 3 * (2 ^ s)
20012 FOR k = 1 TO m
20014 FOR k1 = 1 TO s
20016 tvector(k, k1) = bvector(k, k1): REM reset tvector
20018 NEXT k1
20019 NEXT k

20020 x = 0
20021 FOR k = 1 TO s: REM s terms in vector
20022 tvector(rank(i), k) = INT((RND + .5)): x = x + tvector(rank(i), k)
20030 NEXT k
20032 IF x < 2 THEN GOTO 20020
20100 REM This section must assign the 2fi's with rank(i) term with rank < i term
20110 FOR i1 = 1 TO it
20120 FOR i2 = 1 TO i - 1
20135 IF ita(i1, 1) = rank(i) THEN IF ita(i1, 2) = rank(i2) THEN GOSUB 21000
20140 IF ita(i1, 1) = rank(i2) THEN IF ita(i1, 2) = rank(i) THEN GOSUB 21000

```

```

20150 NEXT i2
20160 NEXT i1
20200 GOSUB 30000: REM need to calculate ztemp
20205 REM WRITE #1, j, ztemp, zcur: REM test code
20210 IF ztemp < zcur THEN GOSUB 32000
20215 IF j = s + 1 THEN GOSUB 32000: REM sets first temp column assignment
20220 NEXT j
20225 RETURN: REM test code
20230 REM now time to allow assignment to any column
20240 FOR j1 = 0 TO 1: REM check all columns
20243 IF j1 + j2 + j3 + j4 + j5 = 5 THEN GOTO 20450: REM 000 not a vector
20245 tvector(rank(i), 1) = 1 - j1
20250 FOR j2 = 0 TO 1
20255 tvector(rank(i), 2) = 1 - j2
20260 FOR j3 = 0 TO 1
20265 tvector(rank(i), 3) = 1 - j3
20266 FOR j4 = 0 TO 1
20267 tvector(rank(i), 4) = 1 - j4
20268 FOR j5 = 0 TO 1
20269 tvector(rank(i), 5) = 1 - j4

20280 IF j1 + j2 + j3 + j5 = 5 THEN GOTO 20420: REM 0000 not a vector
20300 REM This section must assign the 2fi's with rank(i) term with rank <j term
20310 FOR i1 = 1 TO it
20320 FOR i2 = 1 TO i1 - 1
20335 IF ita(i1, 1) = rank(i) THEN IF ita(i1, 2) = rank(i2) THEN GOSUB 22000
20340 IF ita(i1, 1) = rank(i2) THEN IF ita(i1, 2) = rank(i) THEN GOSUB 22000
20350 NEXT i2
20360 NEXT i1
20400 GOSUB 30000: REM need to calculate ztemp
20405 REM WRITE #1, j1, j2, j3, j4, j5, ztemp: REM test code
20410 IF ztemp < zcur THEN GOSUB 32000
20419 NEXT j5
20420 NEXT j4
20430 NEXT j3
20440 NEXT j2
20450 NEXT j1
20500 RETURN

21000 REM assign 2fi
21010 FOR l = 1 TO s
21015 tvector(p + i1, l) = tvector(rank(i), l) + tvector(rank(i2), l)
21020 IF tvector(p + i1, l) = 2 THEN tvector(p + i1, l) = 0: REM mod 2 addition
21080 NEXT l
21090 RETURN

22000 REM assign 2fi
22010 FOR l = 1 TO s
22015 tvector(i1, l) = tvector(rank(i), l) + tvector(rank(i2), l)
22020 IF tvector(i1, l) = 2 THEN tvector(i1, l) = 0: REM mod 2 addition
22080 NEXT l
22090 RETURN

30000 REM this subroutine calculates ztemp for tvector
30005 ztemp = 0
30010 FOR i3 = 1 TO m

```

```

30015 IF tvector(i3, 1) + tvector(i3, 2) + tvector(i3, 3) + tvector(i3, 4) = 0 THEN GOTO 30110
30020 FOR i4 = 1 TO m
30025 IF i3 = i4 THEN GOTO 30100
30030 FOR i5 = 1 TO s
30040 IF tvector(i3, i5) <> tvector(i4, i5) THEN GOTO 30100
30050 NEXT i5
30060 ztemp = ztemp + CI(i3): REM this term is confounded
30070 GOTO 30110: REM only count each ci once
30100 NEXT i4
30110 NEXT i3
30120 RETURN

```

```

32000 REM makes the temp solution the new current solution
32010 ACPTctr = ACPTctr + 1
32110 FOR i3 = 1 TO m
32120 FOR i4 = 1 TO s
32130 cvector(i3, i4) = tvector(i3, i4)
32140 NEXT i4
32150 NEXT i3
32170 zcur = ztemp
32200 RETURN

```

```

33000 REM makes a copy of the old current solution as the basis of the new temp solution the new current
solution
33110 FOR i3 = 1 TO m
33120 FOR i4 = 1 TO s
33130 tvector(i3, i4) = cvector(i3, i4)
33140 NEXT i4
33150 NEXT i3
33370 ztemp = zcur
33300 RETURN

```

```

40000 REM this subroutine calculates zcur for cvector
40005 zcur = 0
40010 FOR i3 = 1 TO m
40020 FOR i4 = 1 TO m
40025 IF i3 = i4 THEN GOTO 40100: REM same factor
40030 FOR i5 = 1 TO s
40040 IF cvector(i3, i5) <> cvector(i4, i5) THEN GOTO 40100
40050 NEXT i5
40060 zcur = zcur + CI(i3): REM this term is confounded
40070 GOTO 40110: REM only count each ci once
40100 NEXT i4
40110 NEXT i3
40120 RETURN

```

```

50000 REM makes the temp solution the new best solution
52110 FOR i3 = 1 TO m
52120 FOR i4 = 1 TO s
52130 bvector(i3, i4) = tvector(i3, i4)
52140 NEXT i4
52150 NEXT i3
52170 zbest = ztemp
52180 IF zbest < .01 THEN GOTO 6000: REM optimal solution found
52190 GOSUB 100000: REM test code

```

52200 RETURN

```
60000 REM this subroutine calculates zbest for bvector
60005 zbest = 0
60010 FOR i3 = 1 TO m
60020 FOR i4 = 1 TO m
60025 IF i3 = i4 THEN GOTO 60100: REM same factor
60030 FOR i5 = 1 TO s
60040 IF bvector(i3, i5) <> bvector(i4, i5) THEN GOTO 60100
60050 NEXT i5
60060 zbest = zbest + CI(i3): REM this term is confounded
60070 GOTO 60110: REM only count each ci once
60100 NEXT i4
60110 NEXT i3
61000 RETURN
```

```
70000 REM this SA subroutine calculates ztemp for tvector
70001 REM WRITE #1, "70000": GOSUB 100000: REM test code
70050 REM this section adjusts the assignment of two-factor interactions
70060 FOR x71 = 1 TO it: REM check each interaction term
70120 FOR l = 1 TO s
70150 tvector(x71 + p, l) = tvector(ita(x71, 1), l) + tvector(ita(x71, 2), l)
70160 IF tvector(x71 + p, l) = 2 THEN tvector(x71 + p, l) = 0: REM mod 2 addition
70165 NEXT l
70170 NEXT x71: REM check all it terms
```

```
71000 REM begin to calculate ztemp
71001 REM WRITE #1, "71000": GOSUB 100000: REM test code
71005 ztemp = 0
71010 FOR i3 = 1 TO m
71015 IF tvector(i3, 1) + tvector(i3, 2) + tvector(i3, 3) + tvector(i3, 4) = 0 THEN ztemp = ztemp + CI(i3): GOTO 71110
71110
71020 FOR i4 = 1 TO m
71025 IF i3 = i4 THEN GOTO 71100
71030 FOR i5 = 1 TO s
71040 IF tvector(i3, i5) <> tvector(i4, i5) THEN GOTO 71100
71050 NEXT i5
71060 ztemp = ztemp + CI(i3): REM this term is confounded
71070 GOTO 71110: REM only count each ci once
71100 NEXT i4
71110 NEXT i3
72000 RETURN
```

```
73000 REM tvector(x1,x2) was changed, check other first-order terms
73001 REM WRITE #1, "73000": GOSUB 100000: REM test code
73010 FOR x73 = 1 TO p
73020 IF x73 = x1 THEN GOTO 73200: REM next x73
73030 FOR x74 = 1 TO s
73050 IF tvector(x73, x74) <> tvector(x1, x2) THEN GOTO 73200: REM next x73
73060 NEXT x74
73070 REM At this point x1 and x73 must be confounded
73080 tvector(x73, x2) = tvector(x73, x2) + 1
73090 IF tvector(x73, x2) > 1.5 THEN tvector(x73, x2) = 0: REM either 0 or 1
73100 REM now x73 and x1 have exchanged column assignments
73110 GOTO 73220
73200 NEXT x73
```

73220 RETURN

100000 REM CLS

106520 WRITE #1, " The assignment found is as follows:"

106540 FOR i = 1 TO m

106543 WRITE #1, "x", i, " tvector is ", tvector(i, 1), tvector(i, 2), tvector(i, 3), tvector(i, 4), tvector(i, 5)

106545 NEXT i

106550 FOR i = 1 TO m

106553 WRITE #1, "x", i, " cvector is ", cvector(i, 1), cvector(i, 2), cvector(i, 3), cvector(i, 4), cvector(i, 5)

106555 NEXT i

106560 FOR i = 1 TO m

106562 WRITE #1, "x", i, " bvector is ", bvector(i, 1), bvector(i, 2), bvector(i, 3), bvector(i, 4), bvector(i, 5)

106569 NEXT i

106590 GOSUB 60000

106595 WRITE #1, "Objective Function Value = ", zbest

106597 WRITE #1, "current solution Value = ", zcur

106599 WRITE #1, "Temp Solution Value = ", ztemp

110000 RETURN: REM test code

Instructions to run the ICAVE heuristic:

The REM statements in the code were designed to document the code and the purpose of each section and line in the code. The code to write a new ICAVE formulation is created by editing the existing lines in the code that defines variables. Edit the code with the specifics of the experiment to be constructed. Specifically:

10 REM Heuristic for solving the DOE/RS problem using greedy heuristic ICAVE

Rewrite the REM statement to identify the problem.

30 n\$ = "cave32l.dat": REM

Change "cave32l.dat" to the filename desired using "filename.extension" format.

60 p = 12: REM How many variables are in the RS?

Change P to the appropriate value.

70 m = 31: REM How many total terms are there in the RS?

Change M to the appropriate value.

80 s = 5: REM What is s, where the experiment size is 2^s

Change S to the appropriate value.

215 Cminab = 10000

Change Cminab to the appropriate value.

```
260 ita(1, 1) = 1: ita(1, 2) = 2: REM ab
261 ita(2, 1) = 3: ita(2, 2) = 4: REM cd
262 ita(3, 1) = 5: ita(3, 2) = 6: REM ef
263 ita(4, 1) = 7: ita(4, 2) = 8: REM gh
264 ita(5, 1) = 9: ita(5, 2) = 10: REM ij
270 ita(6, 1) = 11: ita(6, 2) = 12: REM kl
271 ita(7, 1) = 1: ita(7, 2) = 12: REM al
272 ita(8, 1) = 2: ita(8, 2) = 11: REM bk
273 ita(9, 1) = 3: ita(9, 2) = 10: REM cj
274 ita(10, 1) = 4: ita(10, 2) = 9: REM di
275 ita(11, 1) = 5: ita(11, 2) = 8: REM eh
276 ita(12, 1) = 6: ita(12, 2) = 7: REM fg
277 ita(13, 1) = 2: ita(13, 2) = 7: REM be
280 ita(14, 1) = 1: ita(14, 2) = 4: REM ad
282 ita(15, 1) = 2: ita(15, 2) = 5: REM be
283 ita(16, 1) = 3: ita(16, 2) = 6: REM cf
284 ita(17, 1) = 4: ita(17, 2) = 7: REM dg
285 ita(18, 1) = 1: ita(18, 2) = 5: REM ae
286 ita(19, 1) = 2: ita(19, 2) = 6: REM bf
```

Change lines 260 - 290 to define the interaction terms in the RS.

```
342 IF i < p + 1 THEN x = 100
343 CI(i) = x + i
```

Adjust the weights in this line of code.

```
3000 FOR cycle = 1 TO 100: REM This section assigns first order terms to specific columns
```

Change line 3000 to reflect the maximum number of times you want the code to cycle through the CAVE procedure. Current data indicates that more than 100 cycles will produce diminishing marginal returns.

Save the changes. Compile the code for fast run times if desired. Then run the program. The results of the search will be saved as a text file to "filename.extension."

F. Codes Used to Implement the SA Heuristic

The following is the DESA code. This code was modified for each specific DOE/RS problem and setting of the SA parameters. Line 30 identifies the output file. The information on the RS defined in lines 50 - 400 was changed to reflect the specific terms in the requirement set. The values of *ST*, *CM* and *TL* were changed to correspond to each experimental run. This code was used for experimental design point number six. The problem was the 16/13 DOE/RS problem with the following SA parameter settings: *ST* = 10, *CM* = 0.93, *TL* = 10. Lines 5000 to 5040 contain the inputs for the SA parameters.

```
10 REM Heuristic for solving the DOE/RS problem using simulated annealing
20 REM by Major Steven Forsythe 5 Nov 99
25 RANDOMIZE
30 n$ = "run6a.dat": REM
35 OPEN n$ FOR OUTPUT AS #1
40 CLS
50 WRITE #1, "Solution for DOE/RS Problem"
60 p = 7: REM INPUT "How many first order terms are in the Requirements set?", p
70 m = 13: REM INPUT "how many total terms are there in the RS?", m
80 s = 4: REM INPUT "What is s, where the experiment size is 2^s", s
90 it = m - p: REM it is the number of interaction terms
100 n = 2 ^ s - 1
110 zcur = zbest = ztemp = 100000

200 DIM x(m): REM X(i) is confounded
210 DIM Cl(m): REM c is the weight assigned each term in the RS
215 Cminab = 10000
220 DIM rank(p): REM rank is the ranked order of the first-order terms in RS (highest =1)
225 DIM rkval(p): REM rkval is the sum of the weights of first order terms
228 DIM rnktmp(p): REM temporary values of rkval
230 DIM xvble$(p): REM xvble is the name of the experimental variable
233 DIM tvector(m, s): REM tvector(m,s) is a vector which defines a temporary column assignments
234 DIM cvector(m, s): REM current vector
235 DIM bvector(m, s): REM records the best solution vector so far
240 DIM ita(it, 2): REM This term will define each interaction term
245 DIM d(p, s): REM d is the min aberration set
250 REM define 2fi's
252 ita(1, 1) = 1: ita(1, 2) = 2: REM ab
255 ita(2, 1) = 3: ita(2, 2) = 4: REM cd
260 ita(3, 1) = 5: ita(3, 2) = 6: REM ef
270 ita(4, 1) = 1: ita(4, 2) = 7: REM ag
272 ita(5, 1) = 2: ita(5, 2) = 7: REM bg
273 ita(6, 1) = 3: ita(6, 2) = 6: REM cf
280 d(1, 1) = 1: d(1, 2) = 0: d(1, 3) = 0: d(1, 4) = 0: REM col 1
281 d(2, 1) = 0: d(2, 2) = 1: d(2, 3) = 0: d(2, 4) = 0: REM col 2
```

```

282 d(3, 1) = 0: d(3, 2) = 0: d(3, 3) = 1: d(3, 4) = 0: REM col 4
283 d(4, 1) = 0: d(4, 2) = 0: d(4, 3) = 0: d(4, 4) = 1: REM col 8
284 d(5, 1) = 1: d(5, 2) = 1: d(5, 3) = 1: d(5, 4) = 0: REM col 7
285 d(6, 1) = 0: d(6, 2) = 0: d(6, 3) = 1: d(6, 4) = 1: REM col 12
286 d(7, 1) = 0: d(7, 2) = 1: d(7, 3) = 1: d(7, 4) = 1: REM col 14

300 REM this is the routine to input the RS and the wts for each term in the RS
310 REM write #1, "please enter the names or abbreviations for the experimental variables:"
320 FOR i = 1 TO p
330 xvble$(i) = "X" + STR$(i): REM INPUT xvble$(i)
331 NEXT i
335 FOR i = 1 TO m: REM add the ci's (weights)
341 x = 0
342 IF i < p + 1 THEN x = 100
343 CI(i) = x + i
344 NEXT i
700 REM check input data
710 CLS
720 WRITE #1, "Is the RS as follows?"
725 WRITE #1, "#", "name", "weight"
730 FOR i = 1 TO p
740 WRITE #1, i, xvble$(i), CI(i)
750 NEXT i
760 FOR i = 1 TO it
770 WRITE #1, i + p, ita(i, 1), "**", ita(i, 2), CI(i + p)
780 NEXT i
800 REM INPUT "Is this correct (Y/N)", y$
810 REM IF y$ = "n" THEN STOP
820 REM IF y$ = "N" THEN STOP
830 REM IF y$ = "Y" THEN GOTO 1000
840 REM IF y$ = "y" THEN GOTO 1000
850 REM GOTO 800

1000 REM calculate the most "important variables" and rank order them
1010 REM mkval is the rank values for the r first order terms
1020 REM this loop calculates the sum of the weights associated with a term
1030 FOR i = 1 TO p
1040 mkval(i) = CI(i)
1050 NEXT i
1070 REM now to add the 2it terms to the mkval
1080 FOR i = 1 TO it
1090 FOR k = 1 TO 2
1100 FOR j = 1 TO p
1110 IF ita(i, k) = j THEN mkval(j) = mkval(j) + CI(p + i)
1120 NEXT j
1130 NEXT k
1140 NEXT i: REM this adds wts for the 2it to appropriate mkval's
1150 REM This section creates temporary values to be used to rank order the mkval
1160 FOR i = 1 TO p
1170 mktmp(i) = mkval(i)
1180 NEXT i
1200 REM This section rank orders the first order terms and assigns them to rank(r)
1205 temp = 0
1210 FOR i = 1 TO p: REM i is the ith rank order term
1220 FOR j = 1 TO p: REM j is the jth first order term
1225 IF mktmp(j) > temp THEN temp = mktmp(j): rank(i) = j
1230 NEXT j

```

```

1240 mktmp(rank(i)) = -1: REM this change removes the assigned term from future consideration
1250 temp = 0
1260 NEXT i
2000 REM prints the rank order of the first order terms
2010 WRITE #1, "This is the experimental variables in order of their significance."
2014 WRITE #1, "rank", "variable"
2020 FOR i = 1 TO p
2030 WRITE #1, i, "X", rank(i)
2040 NEXT i
2050 REM INPUT "Continue", x
3000 REM This section assigns first order terms to specific columns
3110 FOR i = 1 TO s
3130 tvector(rank(i), i) = 1: REM The most important term is assigned to (1,0,0...) etc
3135 cvector(rank(i), i) = 1
3140 bvector(rank(i), i) = 1
3150 NEXT i
3200 REM This section must assign the 2fi's
3210 FOR i = 1 TO it
3220 FOR j = 1 TO s
3230 FOR k = 1 TO s
3235 IF ita(i, 1) = rank(j) THEN IF ita(i, 2) = rank(k) THEN GOSUB 10000
3250 NEXT k
3260 NEXT j
3270 NEXT i
3300 REM This section must assign the additional first order terms to columns
3310 FOR i = s + 1 TO p
3320 GOSUB 20000
3325 FOR j1 = 1 TO m
3330 FOR j = 1 TO s
3332 bvector(j1, j) = cvector(j1, j)
3333 NEXT j
3334 NEXT j1
3390 NEXT i
4400 INPUT x: REM test
4500 REM CLS
4520 WRITE #1, "    The assignment found is as follows:"
4540 FOR i = 1 TO p
4560 WRITE #1, xvble$(i), "is assigned to column "
4562 FOR j = 1 TO s
4564 tempcol = tempcol + (2 ^ (j - 1)) * cvector(i, j)
4566 NEXT j
4567 WRITE #1, tempcol: tempcol = 0
4570 NEXT i
4571 FOR i = p + 1 TO m
4572 WRITE #1, "interaction term", i
4573 FOR j = 1 TO s
4574 tempcol = tempcol + (2 ^ (j - 1)) * cvector(i, j)
4576 NEXT j
4577 WRITE #1, tempcol: tempcol = 0
4578 NEXT i
4579 GOSUB 40000
4580 WRITE #1, "Objective Function Value = ", zcur
4600 REM zbest = 1000: zcur = 1000: REM test code

5000 REM ***** SA section *****
5010 REM ***** initialize *****

```

```

5020 stopT = 10: REM stopping temperature
5025 iteration = 0
5030 TL = 10: REM Temperature Length
5040 CM = .93: REM Cooling Multiplier (aka alpha)
5042 WRITE #1, "TL = ", TL, "CM = ", CM
5043 WRITE #1, "Time =", TIMES$
5050 REM calculate starting temp
5060 FOR i = 1 TO m
5070 t = t + CI(i)
5080 NEXT i
5090 startT = t / .693: REM guarantees good acceptance rate
5095 temp = startT
5100 REM temp = 100: REM test code
5110 FOR i1 = 1 TO m
5120 FOR i2 = 1 TO s
5130 bvector(i1, i2) = cvector(i1, i2)
5140 NEXT i2
5150 NEXT i1
5160 zbest = zcur

5200 REM perturb current solution to generate temp solution
5205 iteration = iteration + 1
5210 FOR i1 = 1 TO m
5220 FOR i2 = 1 TO s
5230 tvector(i1, i2) = cvector(i1, i2)
5240 NEXT i2
5250 NEXT i1
5252 REM GOSUB 100000: REM test code
5260 x1 = INT(RND * p) + 1
5270 x2 = INT(RND * s) + 1
5280 IF tvector(x1, x2) = 0 THEN tvector(x1, x2) = 1: GOTO 5300
5290 IF tvector(x1, x2) = 1 THEN tvector(x1, x2) = 0

5300 GOSUB 73000: REM check to see if x1 was confounded with a first-order term
5305 GOSUB 70000: REM calculate ztemp
5307 REM GOSUB 100000: REM test code
5310 IF ztemp < zbest THEN GOSUB 50000
5410 IF ztemp <= zcur THEN GOSUB 32000
5420 IF ztemp > zcur THEN IF RND < EXP((zcur - ztemp) / temp) THEN GOSUB 32000 ELSE GOSUB 33000
5425 IF zbest < .01 THEN GOTO 6000
5430 TLcounter = TLcounter + 1
5440 IF TLcounter > 10 * TL THEN TLcounter = 0: temp = temp * CM: ACPTctr = 0
5442 IF ACPTctr > TL THEN TLcounter = 0: temp = temp * CM: ACPTctr = 0
5445 IF TLcounter = 0 THEN PRINT "Temp: "; temp, "Zcur: "; zcur, "zbest: "; zbest: REM test code"
5447 IF TLcounter = 0 THEN WRITE #1, "Temp:", temp, "Zcur: ", zcur, "zbest: ", zbest: REM test code"
5450 IF temp < stopT THEN GOTO 6000
5500 GOTO 5200
6000 REM CLS
6520 WRITE #1, "    The assignment found is as follows:"
6540 GOSUB 100000
6600 WRITE #1, "Temp = ", temp
6610 WRITE #1, "iteration = ", iteration
6620 WRITE #1, "time = ", TIMES$

```

7000 BEEP: STOP

10000 REM This subroutine assigns columns to the i th $2fi = j*k$ and assign vector

10010 FOR l = 1 TO s

10015 tvector(i + p, l) = tvector(rank(j), l) + tvector(rank(k), l)

10020 IF tvector(i + p, l) = 2 THEN tvector(i + p, l) = 0: REM mod 2 addition

10030 cvector(i + p, l) = tvector(i + p, l)

10040 bvector(i + p, l) = tvector(i + p, l)

10080 NEXT l

10200 RETURN

20000 REM This subroutine assigns s+1 to p first order terms; i = term #

20001 zcur = 0: REM reset current solution on first iteration

20003 FOR j = s + 1 TO p

20012 FOR k = 1 TO m

20014 FOR k1 = 1 TO s

20016 tvector(k, k1) = bvector(k, k1): REM reset tvector

20018 NEXT k1

20019 NEXT k

20020 FOR k = 1 TO s: REM s terms in vector

20022 tvector(rank(i), k) = d(j, k)

20030 NEXT k

20100 REM This section must assign the $2fi$'s with rank(i) term with rank < i term

20110 FOR i1 = 1 TO it

20120 FOR i2 = 1 TO i - 1

20135 IF ita(i1, 1) = rank(i) THEN IF ita(i1, 2) = rank(i2) THEN GOSUB 21000

20140 IF ita(i1, 1) = rank(i2) THEN IF ita(i1, 2) = rank(i) THEN GOSUB 21000

20150 NEXT i2

20160 NEXT i1

20200 GOSUB 30000: REM need to calculate ztemp

20205 WRITE #1, j, ztemp, zcur: REM test code

20210 IF ztemp < zcur THEN GOSUB 32000

20215 IF j = s + 1 THEN GOSUB 32000: REM sets first temp column assignment

20220 NEXT j

20225 RETURN: REM test code

20230 REM now time to allow assignment to any column

20240 FOR j1 = 0 TO 1: REM check all columns

20243 IF j1 + j2 + j3 + j4 = 4 THEN GOTO 20450: REM 000 not a vector

20245 tvector(rank(i), 1) = 1 - j1

20250 FOR j2 = 0 TO 1

20255 tvector(rank(i), 2) = 1 - j2

20260 FOR j3 = 0 TO 1

20265 tvector(rank(i), 3) = 1 - j3

20270 FOR j4 = 0 TO 1

20275 tvector(rank(i), 4) = 1 - j4

20280 IF j1 + j2 + j3 + j4 = 4 THEN GOTO 20420: REM 0000 not a vector

20300 REM This section must assign the $2fi$'s with rank(i) term with rank < j term

20310 FOR i1 = 1 TO it

20320 FOR i2 = 1 TO i - 1

20335 IF ita(i1, 1) = rank(i) THEN IF ita(i1, 2) = rank(i2) THEN GOSUB 22000

20340 IF ita(i1, 1) = rank(i2) THEN IF ita(i1, 2) = rank(i) THEN GOSUB 22000

20350 NEXT i2

20360 NEXT i1

20400 GOSUB 30000: REM need to calculate ztemp

```

20405 WRITE #1, j1, j2, j3, j4, ztemp: REM test code
20410 IF ztemp < zcur THEN GOSUB 32000
20420 NEXT j4
20430 NEXT j3
20440 NEXT j2
20450 NEXT j1
20500 RETURN

```

```

21000 REM assign 2fi
21010 FOR l = 1 TO s
21015 tvector(p + i1, l) = tvector(rank(i), l) + tvector(rank(i2), l)
21020 IF tvector(p + i1, l) = 2 THEN tvector(p + i1, l) = 0: REM mod 2 addition
21080 NEXT l
21090 RETURN

```

```

22000 REM assign 2fi
22010 FOR l = 1 TO s
22015 tvector(i1, l) = tvector(rank(i), l) + tvector(rank(i2), l)
22020 IF tvector(i1, l) = 2 THEN tvector(i1, l) = 0: REM mod 2 addition
22080 NEXT l
22090 RETURN

```

```

30000 REM this subroutine calculates ztemp for tvector
30005 ztemp = 0
30010 FOR i3 = 1 TO m
30015 IF tvector(i3, 1) + tvector(i3, 2) + tvector(i3, 3) + tvector(i3, 4) = 0 THEN GOTO 30110
30020 FOR i4 = 1 TO m
30025 IF i3 = i4 THEN GOTO 30100
30030 FOR i5 = 1 TO s
30040 IF tvector(i3, i5) <> tvector(i4, i5) THEN GOTO 30100
30050 NEXT i5
30060 ztemp = ztemp + CI(i3): REM this term is confounded
30070 GOTO 30110: REM only count each ci once
30100 NEXT i4
30110 NEXT i3
30120 RETURN

```

```

32000 REM makes the temp solution the new current solution
32010 ACPTctr = ACPTctr + 1
32110 FOR i3 = 1 TO m
32120 FOR i4 = 1 TO s
32130 cvector(i3, i4) = tvector(i3, i4)
32140 NEXT i4
32150 NEXT i3
32170 zcur = ztemp
32200 RETURN

```

```

33000 REM makes a copy of the old current solution as the basis of the new temp solution the new current
solution
33110 FOR i3 = 1 TO m
33120 FOR i4 = 1 TO s
33130 tvector(i3, i4) = cvector(i3, i4)
33140 NEXT i4
33150 NEXT i3
33370 ztemp = zcur

```

33300 RETURN

40000 REM this subroutine calculates zcur for cvector
40005 zcur = 0
40010 FOR i3 = 1 TO m
40020 FOR i4 = 1 TO m
40025 IF i3 = i4 THEN GOTO 40100: REM same factor
40030 FOR i5 = 1 TO s
40040 IF cvector(i3, i5) <> cvector(i4, i5) THEN GOTO 40100
40050 NEXT i5
40060 zcur = zcur + CI(i3): REM this term is confounded
40070 GOTO 40110: REM only count each ci once
40100 NEXT i4
40110 NEXT i3
40120 RETURN

50000 REM makes the temp solution the new best solution
52110 FOR i3 = 1 TO m
52120 FOR i4 = 1 TO s
52130 bvector(i3, i4) = tvector(i3, i4)
52140 NEXT i4
52150 NEXT i3
52170 zbest = ztemp
52180 IF zbest < .01 THEN GOTO 6000: REM optimal solution found
52190 GOSUB 100000: REM test code
52200 RETURN

60000 REM this subroutine calculates zbest for bvector
60005 zbest = 0
60010 FOR i3 = 1 TO m
60020 FOR i4 = 1 TO m
60025 IF i3 = i4 THEN GOTO 60100: REM same factor
60030 FOR i5 = 1 TO s
60040 IF bvector(i3, i5) <> bvector(i4, i5) THEN GOTO 60100
60050 NEXT i5
60060 zbest = zbest + CI(i3): REM this term is confounded
60070 GOTO 60110: REM only count each ci once
60100 NEXT i4
60110 NEXT i3
60200 REM need to calculate minimum aberration using minabctr
60205 minabctr = 0
60210 FOR i1 = 1 TO p
60220 FOR i2 = 1 TO p
60230 FOR i3 = 1 TO s
60240 IF bvector(i1, i3) <> d(i2, i3) THEN GOTO 60260
60245 NEXT i3
60250 minabctr = minabctr + 1
60255 GOTO 60280
60260 NEXT i2
60280 NEXT i1
60290 IF minabctr < p THEN zbest = zbest + Cminab
61000 RETURN

70000 REM this SA subroutine calculates ztemp for tvector
70001 REM WRITE #1, "70000": GOSUB 100000: REM test code
70100 REM this section adjusts the assignment of two-factor interactions

```

70110 FOR x71 = 1 TO it: REM check each interaction term
70120 FOR l = 1 TO s
70150 tvector(x71 + p, l) = tvector(ita(x71, 1), l) + tvector(ita(x71, 2), l)
70160 IF tvector(x71 + p, l) = 2 THEN tvector(x71 + p, l) = 0: REM mod 2 addition
70165 NEXT l
70170 NEXT x71: REM check all it terms

71000 REM begin to calculate ztemp
71001 REM WRITE #1, "71000": GOSUB 100000: REM test code
71005 ztemp = 0
71010 FOR i3 = 1 TO m
71015 IF tvector(i3, 1) + tvector(i3, 2) + tvector(i3, 3) + tvector(i3, 4) = 0 THEN ztemp = ztemp + CI(i3): GOTO
71110
71020 FOR i4 = 1 TO m
71025 IF i3 = i4 THEN GOTO 71100
71030 FOR i5 = 1 TO s
71040 IF tvector(i3, i5) <> tvector(i4, i5) THEN GOTO 71100
71050 NEXT i5
71060 ztemp = ztemp + CI(i3): REM this term is confounded
71070 GOTO 71110: REM only count each ci once
71100 NEXT i4
71110 NEXT i3
71200 REM need to calculate minimum aberation using minabctr
71205 minabctr = 0
71210 FOR i1 = 1 TO p
71220 FOR i2 = 1 TO p
71230 FOR i3 = 1 TO s
71240 IF tvector(i1, i3) <> d(i2, i3) THEN GOTO 71260
71245 NEXT i3
71250 minabctr = minabctr + 1
71255 GOTO 71280
71260 NEXT i2
71280 NEXT i1
71290 IF minabctr < p THEN ztemp = ztemp + Cminab
71295 WRITE #1, "minabctr ", minabctr
72000 RETURN

73000 REM tvector(x1,x2) was changed, check other first-order terms
73001 REM WRITE #1, "73000": GOSUB 100000: REM test code
73010 FOR x73 = 1 TO p
73020 IF x73 = x1 THEN GOTO 73200: REM next x73
73030 FOR x74 = 1 TO s
73050 IF tvector(x73, x74) <> tvector(x1, x2) THEN GOTO 73200: REM next x73
73060 NEXT x74
73070 REM At this point x1 and x73 must be confounded
73080 tvector(x73, x2) = tvector(x73, x2) + 1
73090 IF tvector(x73, x2) > 1.5 THEN tvector(x73, x2) = 0: REM either 0 or 1
73100 REM now x73 and x1 have exchanged column assignments
73110 GOTO 73220
73200 NEXT x73
73220 RETURN

100000 REM CLS
106520 WRITE #1, "    The assignment found is as follows:"
106540 FOR i = 1 TO m
106543 WRITE #1, "x", i, " tvector is ", tvector(i, 1), tvector(i, 2), tvector(i, 3), tvector(i, 4)

```

```

106545 NEXT i
106550 FOR i = 1 TO m
106553 WRITE #1, "x", i, " cvector is ", cvector(i, 1), cvector(i, 2), cvector(i, 3), cvector(i, 4)
106555 NEXT i
106560 FOR i = 1 TO m
106562 WRITE #1, "x", i, " bvector is ", bvector(i, 1), bvector(i, 2), bvector(i, 3), bvector(i, 4)
106569 NEXT i

106590 GOSUB 60000
106595 WRITE #1, "Objective Function Value = ", zbest
106597 WRITE #1, "current solution Value = ", zcur
106599 WRITE #1, "Temp Solution Value = ", ztemp
110000 RETURN: REM test code

```

Instructions to run DESA:

The REM statements in the code were designed to document the code and the purpose of each section and line in the code. The code to write a new DESA formulation is created by editing the existing lines in the code that defines variables. Edit the code with the specifics of the experiment to be constructed. Specifically:

10 REM DESA is a simulated annealing approach to solve the 64/63 DOE/RS problem

Rewrite the REM statement to identify the problem

30 n\$ = "run6a.dat": REM

Change "run6a.dat" to the filename desired using "filename.extension" format.

60 p = 7: REM INPUT "How many first order terms are in the Requirements set?", p

Change P to the appropriate value.

70 m = 13: REM INPUT "how many total terms are there in the RS?", m

Change M to the appropriate value.

80 s = 4: REM INPUT "What is s, where the experiment size is 2^s", s

Change S to the appropriate value.

215 Cminab = 10000

Change Cminab to the appropriate value.

```

250 REM define 2fi's
252 ita(1, 1) = 1: ita(1, 2) = 2: REM ab
255 ita(2, 1) = 3: ita(2, 2) = 4: REM cd
260 ita(3, 1) = 5: ita(3, 2) = 6: REM ef
270 ita(4, 1) = 1: ita(4, 2) = 7: REM ag
272 ita(5, 1) = 2: ita(5, 2) = 7: REM bg
273 ita(6, 1) = 3: ita(6, 2) = 6: REM cf

```

Change these lines to define the interaction terms in the RS.

```

280 d(1, 1) = 1: d(1, 2) = 0: d(1, 3) = 0: d(1, 4) = 0: REM col 1
281 d(2, 1) = 0: d(2, 2) = 1: d(2, 3) = 0: d(2, 4) = 0: REM col 2
282 d(3, 1) = 0: d(3, 2) = 0: d(3, 3) = 1: d(3, 4) = 0: REM col 4
283 d(4, 1) = 0: d(4, 2) = 0: d(4, 3) = 0: d(4, 4) = 1: REM col 8
284 d(5, 1) = 1: d(5, 2) = 1: d(5, 3) = 1: d(5, 4) = 0: REM col 7
285 d(6, 1) = 0: d(6, 2) = 0: d(6, 3) = 1: d(6, 4) = 1: REM col 12
286 d(7, 1) = 0: d(7, 2) = 1: d(7, 3) = 1: d(7, 4) = 1: REM col 14

```

If there is a minimim aberation set D, then define the columns in the above lines:

```

343 CI(i) = x + I

```

Adjust the weights in this line of code.

```

5020 stopT = .10: REM stopping temperature

```

Change these lines to define the stopping temperature (stopT). The higher the number, the sooner the search will end.

```

5030 TL = 10: REM Temperature Length

```

Change these lines to define the temperature length (TL). The smaller the number, the sooner the search will end.

```

5040 CM = .96: REM Cooling Multiplier

```

Change these lines to define the cooling multiplier (CM). The smaller the number, the sooner the search will end.

```

5090 startT = t / .693: REM guarantees good acceptance rate

```

If desired, the starting temperature can be specified. Line 5090 automatically scales the starting temperature to a value that guarantees a good acceptance rate but may cause a nearly

random walk behavior early in the search. Lowering the starting temperature will reduce the search time.

Save the changes. Compile the code for fast run times if desired. Then run the program. The results of the search will be saved as a text file to "filename.extension."

G. Cost and Value Functions

Each of the optimization methods used to solve the DOE/RS problem assumes the existence of an additive cost function of the form $\text{cost} = \sum_{i=1}^n C_i x_i$. As first proved by Fishburn (1965), this form of a cost function exists if and only if the attributes are additive independent. Keeney (1992, pp. 133-136) defines four independence conditions: they are preferential, weak-difference, utility, and additive independence. Additive independence is defined as follows:

Additive Independence. Attributes X_1, \dots, X_n are additive independent if the preference order for lotteries does not depend on the joint probability distributions of these lotteries, but depends only on their marginal probability distributions. (Keeney, 1992, p. 134)

As an example of a cost function of 0-1 attributes that does not exhibit additive independence, consider a combat situation where one of the attributes is the structural integrity of a critical bridge. If the combat is going well, a commander might prefer the bridge intact to send troops and supplies across. If the combat is not going well, a commander might prefer the bridge destroyed to prevent the enemy from using it. The value of the bridge is NOT independent of the value of other attributes. In this case, a more complex form of the cost function would need to be used to accurately model the preferences of the decision maker.

In the DOE/RS problem, all the attributes can be represented as zero-one variables. Either a term in the RS is confounded or it is not confounded. Another advantage of this formulation is that confounding is never a good thing in the DOE/RS problem. For any given term in the RS, the experimenter always prefers the term to not be confounded. This preference is independent of the status of any other term in the RS.

The definitions of the other three forms of independence are:

Preferential Independence. The pair of attributes $\{X_1, X_2\}$ are preferentially independent of the other attributes X_3, \dots, X_N , if the preference order for consequences involving only changes in the levels of X_1 and X_2 does not depend on the levels at which attributes X_3, \dots, X_N are fixed.

Weak-Difference Independence. Attribute X_1 is weak-difference independent of attributes X_2, \dots, X_N if the order of preference differences between pairs of X_1 levels does not depend on the levels at which attributes X_2, \dots, X_N are fixed.

Utility Independence. Attribute X_1 is utility independent of attributes X_2, \dots, X_N if the preference order for lotteries involving only changes in the level of X_1 does not depend on the levels at which attributes X_2, \dots, X_N are fixed. (Keeney, 1992, pp. 132-133)

Keeney and Raiffa (1976) and Keeney (1992) provide extensive discussions on the use of cost functions and methods for constructing and testing cost functions.

H. DOE Nomenclature

This section presents the columns of a full factorial design for 16, 32, and 64 design points. Each column is identified by its numeric value, alphabetic nomenclature, and its S-vector representation. Each representation has its own advantages. The numeric value is useful in computer simulations, the alphabetic nomenclature is often used in DOE text and easy to understand, and the S-vector representation allows computer codes to calculate the correct two-factor interaction column associated with any two columns.

“Standard order” is a systematic way to refer to the rows of the design columns (a, b, ab, c, ac, bc, abc, d, ad, ...) Montgomery (1999, p. 295). This research uses a similar approach to refer to the columns of a full factorial design matrix. Table H-1 gives an example of an 8-trial experimental design.

Table H-1: An 8-Trial Full Factorial Design

Column Number	0	1	2	3	4	5	6	7
Column Name	I (the mean)	A	B	AB	C	AC	BC	ABC
Settings for Run 1	+	-	-	+	-	+	+	-
Settings for Run 2	+	+	-	-	-	-	+	+
Settings for Run 3	+	-	+	-	-	+	-	+
Settings for Run 4	+	+	+	+	-	-	-	-
Settings for Run 5	+	-	-	+	+	-	-	+
Settings for Run 6	+	+	-	-	+	+	-	-
Settings for Run 7	+	-	+	-	+	-	+	-
Settings for Run 8	+	+	+	+	+	+	+	+

To construct a 2^{4-1} design, the factor D is usually assigned to column seven. The identity relation is $D=ABC$ or $I=ABCD$. Notice that if column seven is multiplied by itself, ($D * ABC$), then the resulting column is column 0, the identity column. In vector notation, $D = (1,1,1)$ and $ABC = (1,1,1)$. Therefore, $D*ABC = (1+1 \text{ mod } 2, 1+1 \text{ mod } 2, 1+1 \text{ mod } 2)$ which simplifies to $(0,0,0)$. $(0,0,0)$ is the vector notation for the identity column.

Table H-2: A 16-Trial Experimental Design

Column Number	Letter Representation	S-Vector Representation
0	I (the mean)	(0 0 0 0)
1	A	(0 0 0 1)
2	B	(0 0 1 0)
3	AB	(0 0 1 1)
4	C	(0 1 0 0)
5	AC	(0 1 0 1)
6	BC	(0 1 1 0)
7	ABC	(0 1 1 1)
8	D	(1 0 0 0)
9	AD	(1 0 0 1)
10	BD	(1 0 1 0)
11	ABD	(1 0 1 1)
12	CD	(1 1 0 0)
13	ACD	(1 1 0 1)
14	BCD	(1 1 1 0)
15	ABCD	(1 1 1 1)

Table H-3: A 32-Trial Experimental Design

Column Number	Letter Representation	S-Vector Representation
0	I (the mean)	(0 0 0 0 0)
1	A	(0 0 0 0 1)
2	B	(0 0 0 1 0)
3	AB	(0 0 0 1 1)
4	C	(0 0 1 0 0)
5	AC	(0 0 1 0 1)
6	BC	(0 0 1 1 0)
7	ABC	(0 0 1 1 1)
8	D	(0 1 0 0 0)
9	AD	(0 1 0 0 1)
10	BD	(0 1 0 1 0)
11	ABD	(0 1 0 1 1)
12	CD	(0 1 1 0 0)
13	ACD	(0 1 1 0 1)
14	BCD	(0 1 1 1 0)
15	ABCD	(0 1 1 1 1)
16	E	(1 0 0 0 0)
17	AE	(1 0 0 0 1)
18	BE	(1 0 0 1 0)
19	ABE	(1 0 0 1 1)
20	CE	(1 0 1 0 0)
21	ACE	(1 0 1 0 1)
22	BCE	(1 0 1 1 0)
23	ABCE	(1 0 1 1 1)
24	DE	(1 1 0 0 0)
25	ADE	(1 1 0 0 1)
26	BDE	(1 1 0 1 0)
27	ABDE	(1 1 0 1 1)
28	CDE	(1 1 1 0 0)
29	ACDE	(1 1 1 0 1)
30	BCDE	(1 1 1 1 0)
31	ABCDE	(1 1 1 1 1)

Table H-4: A 64-Trial Experimental Design

Column Number	Letter Representation	S-Vector Representation
0	I (the mean)	(0 0 0 0 0 0)
1	A	(0 0 0 0 0 1)
2	B	(0 0 0 0 1 0)
3	AB	(0 0 0 0 1 1)
4	C	(0 0 0 1 0 0)
5	AC	(0 0 0 1 0 1)
6	BC	(0 0 0 1 1 0)
7	ABC	(0 0 0 1 1 1)
8	D	(0 0 1 0 0 0)
9	AD	(0 0 1 0 0 1)
10	BD	(0 0 1 0 1 0)
11	ABD	(0 0 1 0 1 1)
12	CD	(0 0 1 1 0 0)
13	ACD	(0 0 1 1 0 1)
14	BCD	(0 0 1 1 1 0)
15	ABCD	(0 0 1 1 1 1)
16	E	(0 1 0 0 0 0)
17	AE	(0 1 0 0 0 1)
18	BE	(0 1 0 0 1 0)
19	ABE	(0 1 0 0 1 1)
20	CE	(0 1 0 1 0 0)
21	ACE	(0 1 0 1 0 1)
22	BCE	(0 1 0 1 1 0)
23	ABCE	(0 1 0 1 1 1)
24	DE	(0 1 1 0 0 0)
25	ADE	(0 1 1 0 0 1)
26	BDE	(0 1 1 0 1 0)
27	ABDE	(0 1 1 0 1 1)
28	CDE	(0 1 1 1 0 0)
29	ACDE	(0 1 1 1 0 1)
30	BCDE	(0 1 1 1 1 0)
31	ABCDE	(0 1 1 1 1 1)
32	F	(1 0 0 0 0 0)
33	AF	(1 0 0 0 0 1)
34	BF	(1 0 0 0 1 0)
35	ABF	(1 0 0 0 1 1)
36	CF	(1 0 0 1 0 0)
37	ACE	(1 0 0 1 0 1)
38	BCF	(1 0 0 1 1 0)
39	ABCF	(1 0 0 1 1 1)
40	DF	(1 0 1 0 0 0)
41	ADF	(1 0 1 0 0 1)
42	BDF	(1 0 1 0 1 0)
43	ABDF	(1 0 1 0 1 1)
44	CDF	(1 0 1 1 0 0)
45	ACDF	(1 0 1 1 0 1)
46	BCDF	(1 0 1 1 1 0)
47	ABCDF	(1 0 1 1 1 1)
48	EF	(1 1 0 0 0 0)
49	AEF	(1 1 0 0 0 1)
50	BEF	(1 1 0 0 1 0)
51	ABEF	(1 1 0 0 1 1)
52	CEF	(1 1 0 1 0 0)
53	ACEF	(1 1 0 1 0 1)
54	BCEF	(1 1 0 1 1 0)
55	ABCEF	(1 1 0 1 1 1)
56	DEF	(1 1 1 0 0 0)
57	ADEF	(1 1 1 0 0 1)
58	BDEF	(1 1 1 0 1 0)
59	ABDEF	(1 1 1 0 1 1)
60	CDEF	(1 1 1 1 0 0)
61	ACDEF	(1 1 1 1 0 1)
62	BCDEF	(1 1 1 1 1 0)
63	ABCDEF	(1 1 1 1 1 1)

Vita

Major Steven L. Forsythe was born on 10 February, 1962 at Bitburg AB, West Germany. He graduated as the salutatorian from Genoa Area High School in 1980. He received a Bachelor of Science degree in Physics from Kent State University at Kent, Ohio, in 1984, graduating as a member of the Honors College and an honors ROTC graduate. He was then assigned to the Space and Missile Warning Directorate, Strategic Systems, Electronic Systems Division from 1984 until 1989. Positions held during this period included MGS integration Manager, NORAD-OFFUTT Cadre Program Manager, and Chief of Integration for the Computer System Segment Replacement (CSSR) program. He then served at the Philips Laboratory, Kirtland AFB, New Mexico as a High Power Microwave Physicist and a Spacecraft Vulnerability/Survivability Physicist. He earned a Masters Degree in Operations Research from the Air Force Institute of Technology in 1994 and was accepted into the Engineering Ph.D. program at AFIT, specializing in Operations Research. Since 1997, Major Forsythe has been assigned to the Air Force Studies and Analyses Agency at the Pentagon where he is currently Chief, Command and Control Experimentation. He is a graduate of Squadron Officers School and Air Command and Staff College.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 12-12-2000		2. REPORT TYPE		3. DATES COVERED (From - To) 1 JAN 1995 - 30 DEC 2000	
4. TITLE AND SUBTITLE COMPUTER-BASED METHODS FOR CONSTRUCTING TWO-LEVEL FRACTIONAL FACTORIAL EXPERIMENTAL DESIGNS WITH A REQUIREMENT SET				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
				5d. PROJECT NUMBER	
6. AUTHOR(S) Maj Steven L. Forsythe				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENS) 2950 P Street, Building 640 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DSS/ENS/97J-04	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFSAA 1570 Air Force Pentagon Washington DC 20330-1570 <i>Col John Tomick, AFSAA Chief Analyst</i> <i>(703) 588-5936</i> <i>John Tomick@pentagon.af.mil</i>				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>This dissertation developed four methodologies for computer-aided experimental design of two-level fractional factorial designs with requirement sets (DOE/RS). The requirement sets identify all the experimental factors and the appropriate interaction terms to be evaluated in the experiment.</p> <p>The 0-1 IP formulation was developed as an algorithmic approach. The IP formulation provides a standard optimization methodology that guarantees a global optimum solution. Due to the NP-Complete nature of the DOE/RS problem, however, the IP formulation became computationally intractable as the size of the problem increased.</p> <p>Two new heuristics, Column Assignment Via Examination (CAVE) and Iterative Column Assignment Via Examination (ICAVE), were specifically designed to take advantage of the special structure of the fractional factorial design. The CAVE heuristic quickly generates feasible designs that can serve as starting points for other solution techniques. A simulated annealing approach was used as the optimization strategy in the Design of Experiments via Simulated Annealing (DESA) heuristic. DESA performed consistently over a range of SA parameters. Recommended parameter settings were identified.</p> <p>ICAVE provided higher quality solutions at the expense of greater computational effort when compared to the CAVE and DESA heuristics. For smaller DOE/RS problems, the IP formulation generates an optimal solution.</p>					
15. SUBJECT TERMS Design of Experiments, Heuristics, Optimization, Simulated Annealing, Integer Programming					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 189	19a. NAME OF RESPONSIBLE PERSON Dr. Richard F. Deckro
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4325
Standard Form 298 (Rev. 8-98) Prescribed by ANSI Std. Z39-18					Form Approved OMB No. 074-0188